if

CALCULATOARE PERSONALE

Revistă lunară editată de Micro ATCI S.R.L. Tîrgu Mureş

1 / 92
(12)
125 Lei
ISSN 1220-1529

Periferice de intrare

Test: Balanţe contabile

Visual Basic

DR-DOS 6.0 vs MS-DOS5.0

Reţele locale

Memento: Funcţiile Paradox 3.5

Ultima ora!

S-a infiintat POLICOM S.A. *, societate comerciala multinationala cu capital privat, avind principal actionar Ab Mod* S.N.C

Noua firma va ofera un sortiment larg de produse :

- *Calculatoare personale compatibile* IBM
- *Imprimante matriciale*
- *Dispozitive grafice*
- *Floppy - dischete*
- *Produse software originale*
- *Copiatoare*
- *Truse si accesorii pentru intretinere*

Nu uitati !

*POLICOM*S.A. *un partener potential pentru afacerile dumneavoastra*

CLUJ : B-dul 22 Decembrie nr. 135 tel. 95-156350

if

revistă de informatică
editată de firma **Micro ATCI**

Director: ing. Dumitru Dunca

Redacţia:
ing. Iosif Fettich,
ing. Ingrid Maier,
ing. Romulus Maier,

La acest număr au colaborat:
ing. Vasile Brînduş
mat. Nicolae Galfi
ing. Răzvan Grigorescu,
mat. Vasile.Man
ing. Bogdan Munteanu
fiz. Vera Radovici
mat. Eugen Rotaru
ing. Răzvan Sandu
stud. Socaciu Ioan-Tiberiu
ing. Marius Sturzoiu

Tiparul: tipografia Tîrgu Mureş

Revista apare lunar.

Preţ: 125 lei

Manuscrise originale sau listinguri de programe sînt primite cu plăcere de redacţie, cu condiţia să nu fi fost publicate şi în altă parte. Prin expedierea unui manuscris pe adresa redacţiei, autorul consimte implicit la publicarea materialului său în cadrul revistei. Onorariul se negociază cu redacţia. Materialele nepublicate nu se înapoiază şi nu se reţin.

Revista noastră vă oferă spaţiu pentru reclamă şi publicitate. Pentru amănunte vă rugăm să luaţi legătura cu redacţia.

Cei care doresc să anexeze revistei pliante publicitare tipărite în regie proprie, sînt rugaţi şi ei să se adreseze redacţiei .

Adresa şi telefonul redacţiei:
Micro ATCI, redacţia **"if"**,
RO-4300 Tîrgu Mureş,
C.P. 172, O.P. 1,
tel./fax 954/31660 (direct),
33612, 24158, 20057, 33511,
int. 134 sau 189
41417 (după ora 19)
telex 65354 (Intertours)

Ofuri

"**if** ist nicht tot, Umwertung aller Werte" ar zice Nietzsche dac-ar mai trăi. Dar el nu mai e şi nice **if** nu se simte prea bine. Altfel ierarhiile se schimbă, pînă mai deunăzi un număr costa cam cît 2,5 l de benzină, iar azi... Aceasta nu înseamnă însă că revista noastră s-ar fi ieftinit, deşi ne-ar fi plăcut să fie aşa

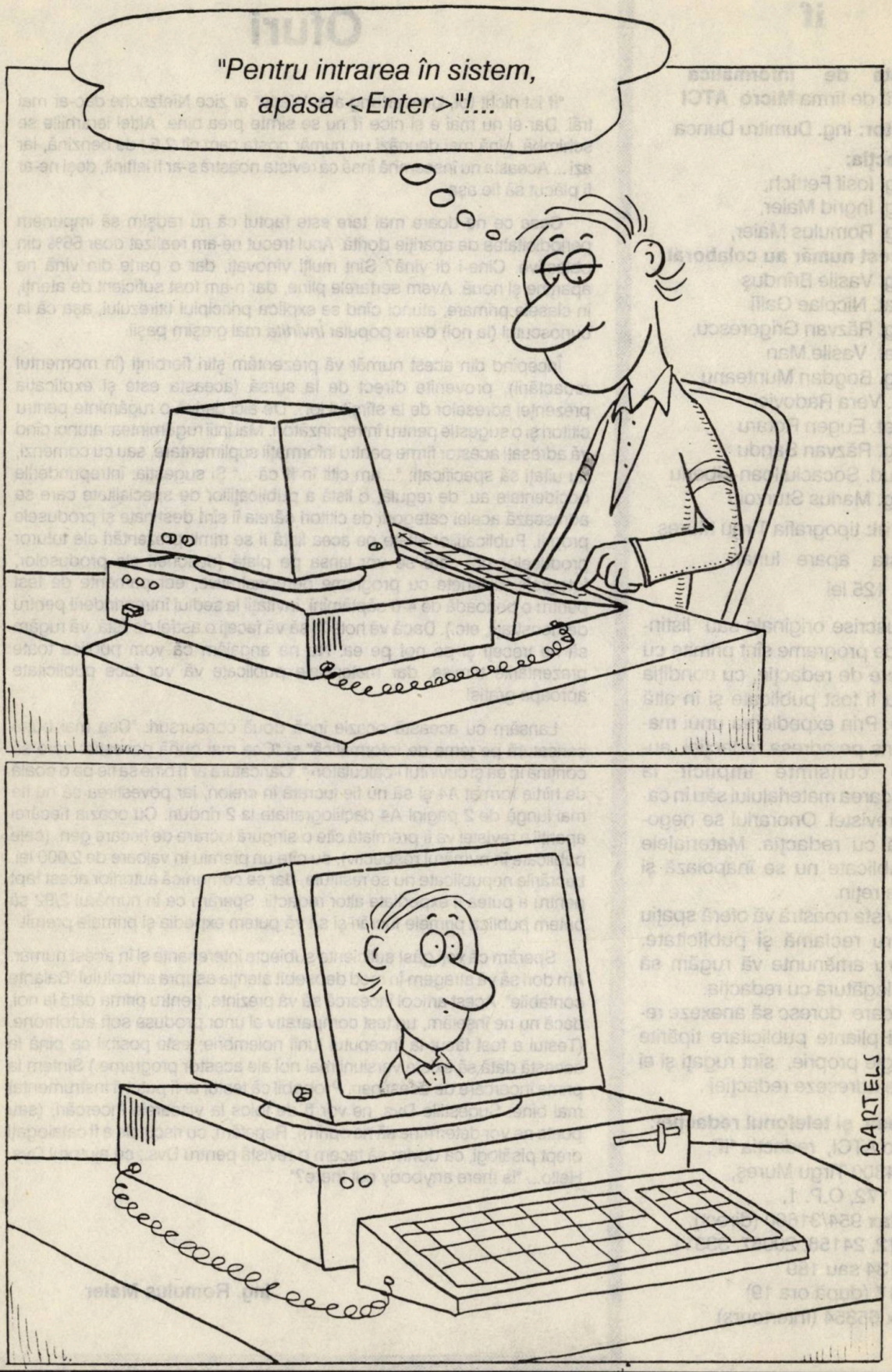
Ceea ce ne doare mai tare este faptul că nu reuşim să impunem periodicitatea de apariţie dorită. Anul trecut ne-am realizat doar 66% din obiective. Cine-i di vină? Sînt mulţi vinovaţi, dar o parte din vină ne aparţine şi nouă. Avem sertarele pline, dar n-am fost suficient de atenţi, în clasele primare, atunci cînd se explica principiul titirezului, aşa că la cunoscutul (la noi) dans popular *Învîrtita* mai greşim paşii.

Începînd din acest număr vă prezentăm ştiri fierbinţi (în momentul redactării), provenite direct de la sursă (aceasta este şi explicaţia prezenţei adreselor de la sfîrşitul lor). De aici derivă o rugăminte pentru cititori şi o sugestie pentru întreprinzători. Mai întîi rugămintea: atunci cînd vă adresaţi acestor firme pentru informaţii suplimentare, sau cu comenzi, nu uitaţi să specificaţi: "...am citit în **if** că ..." Şi sugestia: întreprinderile occidentale au, de regulă, o listă a publicaţiilor de specialitate care se adresează acelei categorii de cititori căreia îi sînt destinate şi produsele proprii. Publicaţiilor aflate pe acea listă li se trimit prezentări ale tuturor produselor noi care se vor lansa pe piaţă (descrieri ale produselor, fotografii, dischete cu programe demonstrative, echipamente de test pentru o perioadă de 4-6 săptămîni, invitaţii la sediul întreprinderii pentru demonstraţii, etc.). Dacă vă hotărîţi să vă faceţi o astfel de listă, vă rugăm să ne treceţi şi pe noi pe ea. Nu ne angajăm că vom publica toate prezentările trimise, dar materialele publicate vă vor face publicitate aproape gratis!

Lansăm cu această ocazie încă două concursuri: "Cea mai bună caricatură pe teme de informatică" şi "Cea mai bună povestire care să conţină în ea şi cuvîntul «calculator»". Caricatura ar fi bine să fie pe o coală de hîrtie format A4 şi să nu fie lucrată în creion, iar povestirea să nu fie mai lungă de 2 pagini A4 dactilografiate la 2 rînduri. Cu ocazia fiecărei apariţii a revistei va fi premiată cîte o singură lucrare de fiecare gen, (cele publicate în numărul respectiv), cu cîte un premiu în valoare de 2.000 lei. Lucrările nepublicate nu se restituie, dar se comunică autorilor acest fapt pentru a putea fi expediate altor redacţii. Sperăm ca în numărul 2/92 să putem publica primele lucrări şi să vă putem expedia şi primele premii.

Sperăm că veţi găsi suficiente subiecte interesante şi în acest număr. Am dori să vă atragem în mod deosebit atenţia asupra articolului "Balanţe contabile". Acest articol încearcă să vă prezinte, pentru prima dată la noi, dacă nu ne înşelăm, un test comparativ al unor produse soft autohtone. (Testul a fost făcut la începutul lunii noiembrie; este posibil ca pînă la această dată să existe versiuni mai noi ale acestor programe.) Sîntem la prima încercare de acest gen. Probabil că testul ar fi putut fi instrumentat mai bine. Sugestiile Dvs. ne vor fi de folos la viitoarele încercări, (sau poate ne vor determina să ne oprim). Repetăm, cu riscul de a fi catalogaţi drept pisălogi, că dorim să facem o revistă pentru Dvs., cu ajutorul Dvs. Hello... "Is there anybody out there?"

ing. Romulus Maier



Cartoon: Joachim Bartels

Cuprins

Expoziţii

- Systems '91 . . . . . pag. 4
- Comdex '91 . . . . . pag. 4
- Info-Market '91 . . . . . pag. 4

Ştiri

- FotoMan . . . . . pag. 5
- CATD - Computer Added Textil Design pag. 6
- WordPerfect 1.0 pentru Windows . . . pag. 6
- Listare color în format de ziar . . . . . pag. 7
- Virus - Utilities . . . . . pag. 7

Sisteme de operare

DR-DOS 6.0 vs MS-DOS 5.0 . . . . . pag. 8

Novell cumpără Digital Research . . . . pag. 11

Profil

Le roi est mort! Vive le roi! pag.10

Portret Philippe Kahn

Test

Balanţe contabile pag.12

- BALPRO . . . . . pag. 13
- BAL-CON . . . . . pag. 13
- CONTAS . . . . . pag. 14
- BALL . . . . . pag. 14

Periferice de intrare

Periferice de intrare . . . . . pag. 16

Tastatura, ligthpen, mouse, joystick,tableta grafică, trackball, touchmouse, scanner, ...

Touch me, user! . . . . . pag. 22

Cîte ceva despre cum lucrează un touchscreen

Vă prezentăm

Visual Basic - programare intuitivă pag.24

Inteligenţa artificială

Ce este bioinformatica? pag.25

Reţele locale

Programele de educaţie Novell . . . . . pag. 31

Reţele locale sau mainframe ?
- o analiză a costurilor "la bani mărunţi" pag. 32

Baze de date

Servere de baze de date . . . . . pag. 37

- Emerald Bay 2.27 (RSPI) . . . . . pag. 37
- Ingres for OS/2, DOS (Ingres) . . . . . pag. 37
- Microsoft SQL Server 1.1 (Microsoft) . pag. 37
- NetWare SQL 2.11 (Novell) . . . . . pag. 37
- Oracle Server 6.0 (Oracle Corp.) . . . pag. 38
- SQL Base Server 4.1 (Gupta Tech.) . . pag. 38

Fundamente

MS-DOS:
Sistemul de gestiune al fişierelor pag.39

Cursuri

Turbo Prolog (partea a treia) pag.43

OOP: C++ - (partea a doua) pag.46

Practică

- 1024 . . . . . pag.52
- Parcarea harddisk-urilor pag.52
- Ce zi e astăzi? pag.52
- Program cu repetiţie pag.53
- Subdirectoare pag.53

Rubrici

- Caseta redacţiei . . . . . pag. 1
- Editorial . . . . . pag. 1
- Colţul programatorului pag.29
- Poşta redacţiei pag.54
- Mica publicitate pag.54

DARIAN ROM SUISSE Srl

Distributor :

ROM CONTROL DATA
KT TEHNOLOGY PTE
FOURTH SHIFT CORPORATION
RING COMPUTER

Va pune la dispozitie programul de amnistie :

FOXPRO 2.0

la urmatoarele preturi :

FOXPRO 2 0 MULTI USER(nelimitat)....	275.000	Lei
FOXPRO 2.0 SINGLE USER	215.000	Lei
DISTRIBUTION KIT (RUNTIME)	175.000	Lei
FOXGRAPH.....	185.000	Lei

Servicii hardware :

- ☑ *calculatoare compatibile IBM PC*
- ☑ *retele NOVELL*
- ☑ *garantie, postgarantie*

Servicii software si consulting :

- ☑ *analiza sitemului informational*
- ☑ *implementarea sistemului integrat de gestiune urmarire si conducere a intreprinderii*

FOURTH SHIFT

- ☑ *cursuri de initiere si perfectionare*

Relatii suplimentare la sediul firmei noastre :

CLUJ, OBSERVATORULUI 146/15
TEL : 95-123611 FAX : 95-124567

FotoMan - aparat de fotografiat alb-negru pentru PC-uri

FotoMan este o cameră digitală concepută în mod special pentru PC-uri. În comparaţie cu o cameră video analogică FotoMan nu necesită nici cartelă Frame-Grabber suplimentară şi nici cartelă de digitalizare şi deci nu are nevoie nici de o mufă suplimentară în PC deoarece FotoMan lucrează în mod digital. Imaginile în nuanţe de gri pot fi transferate direct în PC, prin interfaţa serială. Un chip de memorie RAM intern memorează pînă la 32 de imagini şi le transmite apoi software-ului, imaginile rezultate fiind de foarte bună calitate. În acest caz nu mai apar pierderi de calitate ca în cazul transformării imaginilor analogice în imagini digitale.

Bateriile incorporate permit o funcţionare independentă de reţea de 24 de ore. Ele se reîncarcă automat atunci cînd aparatul este aşezat în unitatea de bază (prin intermediul căreia sînt trimise şi imaginile în calculator). Încărcarea nu durează mai mult de 6 secunde şi poate fi realizată şi de la bateria maşinii.

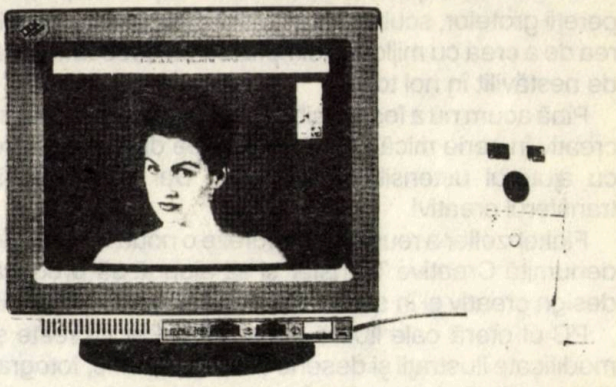
Semnale sonore specifice semnalizează din timp necesitatea reîncărcării bateriei, umplerea memoriei (cu cele 32 de poziţii) sau faptul că aparatul este gata de lucru pentru următoarea fotografie.

După ce cele 32 de înregistrări au fost făcute, FotoMan poate fi cuplat printr-un cablu de legătură normal la interfaţa serială a PC-ului. Cu ajutorul soft-ului FotoTouch, dezvoltat de Logitech în mod special, imaginile pot fi încărcate în PC şi pot fi memorate, prelucrate sau şterse. FotoTouch goleşte în mod automat memoria de imagini a camerei FotoMan, şi ea devine din nou aptă pentru 32 de înregistrări.

Camera digitală poate fi cuplată prin interfaţa serială cu orice PC portabil, indiferent că e un Laptop sau Notebook. La cîteva secunde după înregistrarea imaginii ea poate fi vizualizată pe Notebook cu ajutorul programului FotoTouch, poate fi prelucrată sau poate fi expediată via Modem.

Un bliţ înglobat garantează imagini iluminate optim şi în încăperi întunecoase. Timpii de expunere sînt de 1/25 secunde cu bliţ şi de la 1/1000 pînă la 1/30 secunde în mediu luminos. Atunci cînd este necesar bliţul se declanşează automat. Avînd o greutate de doar 280 g şi dimensiuni reduse de 16,8 cm x 8,1 cm x 3 cm camera poate fi pusă comod într-un buzunar. Datorită unor adîncituri ergonomice pentru degete atît stîngacii cît şi dreptacii pot fotografia cu FotoMan fără balansări, în momentul declanşării. Lentila standard corespunde unei lungimi de focalizare de 55 mm într-un format de 35 mm.

La o cameră normală aceasta ar corespunde la o diafragmă f4,5. Dacă FotoMan ar lucra cu filtru, atunci sensibilitatea acestuia ar trebui să fie de cca. ASA 200. Lentila este prevăzută cu un filet, care permite montarea unei alte lentile sau a unui filtru. Distanţa la fotografiere cu FotoMan este de la 1 metru (cu bliţ 1,2 metri) la infinit.



Rezoluţia unei imagini este de 376x282 pixeli, ceea ce corespunde la o rezoluţie de 75 dpi, calitatea pozelor fiind suficientă pentru ziare, broşuri, etc. O poză memorată necesită cca. 100 KB.

FotoTouch este o dezvoltare a software-ului Ansel conceput pentru scanner-ele Scanan 256 şi permite prelucrarea imaginilor în 256 nuanţe de gri. FotoTouch recunoaşte automat dacă FotoMan se găseşte sau nu în unitatea de bază şi selectează automat viteza maximă de transfer pentru preluarea fotografiilor. Fotografiile aflate în memoria lui FotoMan sînt afişate în aşa-numitul mod previzualizare, în blocuri de cîte 5, de mărimea unor timbre. Cu ajutorul mouse-ului se selectează fotografiile dorite, acestea fiind apoi memorate în PC. Se poate specifica numele fişierului în care să se facă memorarea (max. 8 caractere), dacă se doreşte se poate înregistra şi numele fotografului. Imaginile pot fi memorate în format TIFF, PCX, BMP sau EPS.

Software-ul permite şi luarea unor instantanee la "locul de muncă", putîndu-se specifica intervalul de timp după care să se facă declanşarea automată. FotoTouch permite şi o prelucrare a imaginilor obţinute, cum ar fi mărirea/micşorarea luminozităţii, mărirea/micşorarea imaginii, decupări, rotiri sau scalări ale acesteia, etc.

FotoMan şi FotoTouch lucrează cu orice PC-uri de tip IBM 286, 386 sau 486, sisteme PS/2 şi compatibile. Se presupune o memorie minimă de lucru de 1 MB RAM, dar se recomandă 2 MB. Software-ul lucrează sub Microsoft Windows 3.0. FotoMan poate fi utilizat cu orice cartelă grafică. EGA, VGA, Super VGA, Hercules şi compatibile. Deoarece camera digitală se cuplează serial, la PC trebuie să existe o interfaţă RS-232 liberă.

FotoMan va putea fi obţinut începînd cu sfîrşitul lunii ianuarie 1992 la toate magazinele de specialitate, preţul fiind de cca. 1649 DM (inclusiv impozitul pe valoarea adăugată).

Logitech SA
CH - 1122 Romanel/Morges, Switzerland
Tel: 41(0)218699656, Fax: 41(0)218699717

Computer Added Textil Design (CATD)

Încă de cînd primii oameni au început să picteze pereţii grotelor, sculele lor şi îmbrăcămintea lor, pornirea de a crea cu mijloace simple lucruri deosebite este de nestăvilit în noi toţi.

Pînă acum nu a fost posibilă executarea unui design creativ în serie mică sau de unicat pe diferitele textile cu ajutorul ustensilelor existente; Dar acum există transferul creativ!

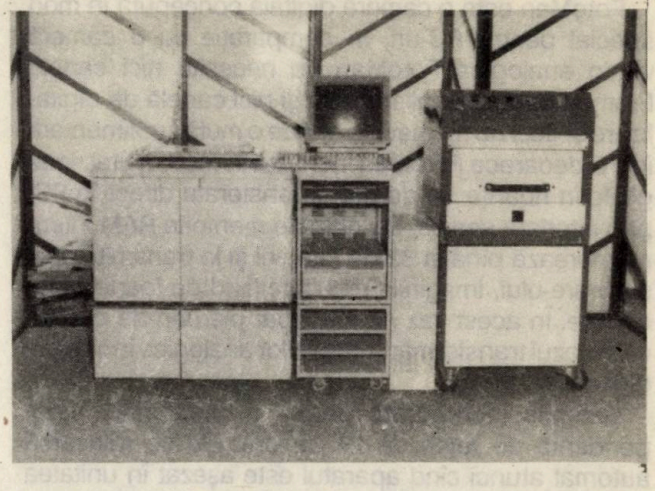
Finkenzeller a reuşit să elaboreze o nouă tehnologie denumită Creative Transfer şi să ajungă să producă design creativ şi în serie mică, ieftin, repede şi flexibil.

PC-ul oferă cale liberă creativităţii. Pot fi create şi modificate ilustraţii şi desene de toate tipurile, fotografii şi inscripţii cît şi grafice în toate variaţiunile.

Cu ajutorul imprimantei laser color cuplate la PC aceste proiecte şi desene pot fi transpuse în doar cîteva secunde pe hîrtie. Pagina astfel obţinută, tratată cu un fluid special, se poziţionează pe materialul textil pregătit în presa necesară. Toate graficele şi toate imaginile vor fi impregnate în materialul textil la o temperatură de 210 grade şi o presiune de peste 2 tone.

Dezvoltarea acestui sistem este practic nelimitată, fiind posibilă captarea imaginilor video, a diapozitivelor, a fotografiilor precum şi prelucrarea grafică a tuturor informaţiilor.

În timpul testului la care am asistat, în mai puţin de 5 minute o imagine color prelucrată pe PC a fost



transpusă pe hîrtie şi de acolo pe un tricou de bumbac. Culoarea se impregnează în material şi nu se aplică pe acesta, ceea ce înseamnă că la întindere imaginea nu crapă. Un material textil astfel impregnat poate fi spălat la o temperatură de 40-50 grade Celsius, fără ca imaginea să aibă de suferit. Este posibilă şi impregnarea de imagini pe sticlă, materiale plastice, metale. O instalaţie completă costă în jur de 80.000 DM.

Carl Finkenzeller GmbH
Ingolstädter Str. 61n, 8000 München 45
Tel. 089/31818 - 103, Fax: 31818 - 138

WordPerfect 1.0 pentru Windows

La Systems '91 a fost prezentată prima versiune a procesorului de texte WordPerfect pentru Windows. Cu aceasta renumitul procesor de texte este disponibil nu doar pe DOS-PC-uri, Mac-uri, pe calculatoarele Unix şi Next, pe VAX, calculatoarele mari IBM şi pe sistemele Data General ci şi sub Windows 3.0.

Paleta de funcţii a lui WordPerfect 1.0 se bazează pe funcţiile versiunii DOS WordPerfect 5.1, fiind extinsă cu facilităţi suplimentare care utilizează avantajele lui Windows 3.0. Programul lucrează în modul WYSIWYG, ceea ce înseamnă că toate formatările de text, fonturile, graficele, tabelele, notele de subsol şi formulele unui document vor avea o reprezentare identică pe ecran şi la imprimantă.

Prin intermediul noilor funcţii tabulatorii, marginile paginii, coloanele, tabelele şi distanţele dintre rînduri pot fi definite direct pe ecran cu ajutorul mouse-ului. Toate funcţiile pot fi apelate, în funcţie de preferinţe, cu ajutorul mouse-ului, prin intermediul meniurilor pull-down, sau de la tastatură.

Cu totul nouă este funcţia WYSBYGI (What You See Before You Get It). În modul previzualizare documentele dorite din director pot fi vizualizate într-o fereastră pe monitor şi pot fi apoi încărcate. Această reprezentare a formatării exacte cu toate atributele textului are loc la un simplu clic pe mouse.

Funcţiile utilizate mai frecvent şi macrourile dintr-un meniu pot fi aranjate liber într-o listă de simboluri, denumită "Button Bar", reprezentate grafic şi activate apoi printr-o tastare pe mouse.

Programul permite cuplarea şi prelucrarea graficelor, dispune de un editor de formule, de generatoare de tabele şi de formulare, de stiluri ca şi de gestiunea notelor de subsol şi de sfîrşit. WordPerfect 1.0 pentru Windows este compatibil cu versiunea DOS WordPerfect 5.1, ceea ce înseamnă că între cele două versiuni pot fi schimbate texte fără probleme.

Preţul produsului este de cca. 1390 DM. Costul trecerii de la WordPerfect 5.1 la WordPerfect 1.0 pentru Windows este de cca. 350 DM.

Ca o curiozitate să amintim faptul că pînă în luna august a acestui an, numai în SUA au fost livrate mai mult de 2 milioane de versiuni. (În tot anul 1990 au fost livrate 2,35 milioane versiuni WordPerfect 5.1). Pentru comparaţie, în 1990 au fost vîndute 1,6 milioane de aplicaţii Windows.

WordPerfect Software GmbH
Frankfurter Strasse 21-25, 6236 Eschborn
Tel. (06196) 904 - 01, Fax (06196) 46003

Listare color în format de ziar

Producătorul de imprimante Tektronix a lansat pe piaţă prima imprimantă color de 300 dpi, numită "PhaserJet PXi" şi care poate tipări în format de ziar aproape orice tip de hîrtie.

Tipăririle pe o dimensiune a hîrtiei de 297 x 446 mm dau utilizatorului posibilitatea de a face reprezentări în dimensiuni realiste. El poate alege dintr-un set de peste 16 milioane de nuanţe de culori.

Imprimanta PhaserJet PXi utilizează limbajul Postscript-Level 2 al firmei Adobe Systems care a îmbunătăţit performanţele standardului Postscript cu cîteva caracteristici şi care oferă de exemplu o adaptare a culorilor independentă de aparatură.

Prelucrarea extrem de rapidă a fişierelor Postscript se datorează procesorului RISC de 24 MHz înglobat în PhaserJet PXi. Paginile color sînt listate în doar 2 minute, în timp ce cele alb-negru se pot lista în 40-60 secunde, în funcţie de densitatea textului.

Configuraţia de bază a imprimantei PhaserJet PXi constă dintr-o memorie de 10 MByte, care poate fi extinsă, cu de două ori 4 MByte, pînă la 18 MByte. Prima extensie de 4 MByte oferă suficientă memorie pentru listarea unei imagini de dimensiuni maxime (297 x 446 mm), cea de-a doua extensie suplimentează memoria penrtu buffer-ul de intrare, seturile de caractere şi prelucrările de imagini în timpul listării.

Imprimanta PhaserJet PXi poate fi legată la mai multe calculatoare şi în reţea. Furnitura de livrare conţine şi un conector Ethernet.

Preţ: 37.000 DM. Imprimanta a fost expusă atît la TIB '91 (standul Xerox, pavilion F) cît şi la SYSTEMS '91.

Tektronix CmbH
Colonia-Allee 11
D-W 5000 Koln 80,
Tel: (0221) 96969 - 0

Virus - Utilities

Firma Ikarus Software a prezentat la Systems, în München trei noi produse: Virus - Utilities Advanced Edition, Virus - Utilities for Windows şi programul interactiv de învăţare ViruSchool.

Ceea ce ne-a determinat să vă prezentăm produsele acestei firme este maniera nouă de abordare a problemei: a fost elaborat un limbaj de descriere a viruşilor, denumit VDL (Virus Description Language), care permite descrierea noilor soiuri de bestii apărute, preluarea acestora în baza de cunoştinţe a sistemului, protecţia putînd fi extinsă şi asupra acestora. Avantajele ce derivă din folosirea unui limbaj de descriere a viruşilor ni se par deosebite: cei interesaţi, "epidemiologii" calculatoarelor, nu vor mai trebui să schimbe între ei viruşi adevăraţi, potenţial periculoşi, ci doar informaţii descriptive suficiente pentru combaterea eficientă a lor.

Versiunea 1.5. recunoaşte peste 750 de viruşi, permite calcularea sumelor de control, a supravegherii sistemului pentru detectarea activităţii unor posibili viruşi şi permite refacerea fişierelor infectate chiar şi în cazul viruşilor care rescriu informaţiile din fişier - datele de configuraţie, sistem şi secţiunile program pasibile de a fi rescrise putînd fi salvate, pentru siguranţă, pe o dischetă. Protecţia este asigurată şi sub Windows. ViruSchool este un program şcoală care permite utilizatorului să înveţe, conversînd cu calculatorul, ce este un virus, cum se poate preveni infectarea calculatorului şi ce se poate face atunci cînd calculatorul s-a îmbolnăvit. Un pachet complet costă 565 DM.

Firma Ikarus Software editează lunar foaia informativă "Ikarus Hotline", în care sînt prezentate informaţii la zi din lumea viruşilor şi a protecţiei datelor, publicaţia fiind

distribuită gratuit utilizatorilor înregistraţi ai programului.

Ikarus Software Gmbh,
Franz Josef str. 7,
5700 Zell am See, Austria,
tel. 0043(0)6542/2941-0
fax: 0043(0)6542/2941-15

DR-DOS 6.0 vs MS -DOS 5.0

Cînd în iunie 1990 Digital Research a scos primul sistem de operare DOS cu un management de memorie îmbunătăţit, Microsoft s-a grăbit să anunţe MS-DOS 5.0, care a ieşit în lume în iulie '91. Dar prea mult timp de odihnă nu le rămînea soft-iştilor din Redmond - cu DR-DOS 6.0, Digital Research încearcă din nou să se distanţeze.

MS-DOS foloseşte driver-ul "Himem.sys" pentru administrarea memoriei. Pe PC-uri cu 386 sau 486, "emm386.exe" are grijă de cererea de UMB-uri ("upper memory blocks") şi, la dorinţă, de "transformarea" memoriei XMS în EMS (extinsă în expandată). În directorul rădăcină se mai găseşte misteriosul, dar pentru Windows indispensabilul, fişier "wina20.386".

DR-DOS oferă "hidos.sys" pentru 286, respectiv "emm386.sys" pentru 386. Driverele oferite de DR-DOS sînt mai eficiente decît cele oferite de Microsoft: "hidos.sys" lucrează direct cu mai multe seturi de cipuri (NEAT, LEAP, SCAT) şi permite străpungerea barierei celor 600 kocteţi chiar şi pe XT-uri dotate cu EMS (pe AT-uri 286, DR-DOS lasă disponibili maxim 628 kocteţi, MS-DOS 621 kocteţi), iar "emm 386.sys" ştie să folosească şi memoria video şi să copieze ROM în RAM.

Ambele drivere 386 comută procesorul în modul virtual, ceea ce generează probleme în cazul unor programe (de exemplu, cele ce lucrează cu DOS-extenderul de la PharLap, cum ar fi "Mathematica"). Nici Windows nu se mai poate utiliza în modul standard - "emm386.sys" din DR-DOS are pentru aceasta comutatorul "/winstd", care permite apoi lansarea Windows în toate cele 3 moduri.

UMB-urile create se pot folosi prin intermediul comenzilor "devicehigh" în config.sys şi "loadhigh" în autoexec.bat. Astfel, drivere şi TSR-uri sînt încărcate în UMB, nemaiocupînd memorie "standard", convenţională. La toate driverele şi utilitarele care fac parte din furnitură, acest lucru funcţionează fără probleme, dar la utilitare oferite de terţi, instalarea poate deveni delicată. Căci TSR-uri care după lansare "se umflă" sau "slăbesc" (ca de exemplu Shell şi Desktop din PC-Tools) nu se prea împacă cu modul de administrare al memoriei propus.

Memoria suplimentară, fie ea extinsă sau expandată, este folosită direct de unele programe, dar, de regulă, cel mai bun lucru este ca ea să fie folosită de un cache de harddisk. Dacă aici Microsoft-ul se prezintă cu deja cunoscutul Smartdrive, Digital Research vine cu unul din cele mai performante programe cache disponibile, Super PC-Kwik al firmei Multisoft. În timp ce Smartdrive se fixează în memoria extinsă şi mai eliberează din memoria ocupată numai în folosul Windows-ului, PC-Kwik îşi "adună" singur memoria de lucru şi, la cerere, o cedează altor programe. Un driver suplimentar asigură compatibilitatea cu modurile "standard" şi "enhanced" pentru Windows. Ambele programe cache "grăbesc" mult harddisk-ul.

Din cele amintite pînă aici este evident că instalarea noilor sisteme de operare nu este o treabă banală. Acesta este şi motivul pentru care ambele sisteme de operare sînt însoţite de programe de instalare puternice.

DR-DOS oferă 3 opţiuni: multă "funcţionalitate", multă memorie, sau calea de mijloc. Cum "funcţionalitatea" depinde însă prea ades de utilizarea particulară a PC-ului şi nu întotdeauna se potriveşte cu valorile implicite fixate de DR-DOS, nu poate fi evitată setarea individuală a parametrilor sistemului. Lucru care se întîmplă interactiv, prin intermediul unui help online. Din păcate, deseori programul de instalare nu urmăreşte setările făcute, astfel încît opţiuni care se exclud de fapt generează confuzii (ducînd, pînă la urmă, la consultarea manualului...).

MS-DOS lucrează altfel: trebuie răspuns la cîteva întrebări, după care programul de instalare construieşte o configuraţie standard. Cine vrea să-şi acordeze sistemul mai fin, e trimis direct la manualul de utilizare.

Ambele sisteme permit stocarea vechiului sistem de operare pe harddisk şi revenirea la el, în cazul că se doreşte acest lucru. DR-DOS are pentru aceasta comanda "uninstall", în timp ce MS-DOS creează la instalare o dischetă boot-abilă ce rezolvă acelaşi lucru.

Ambele sisteme dispun de o suprafaţă utilizator grafică, aşa-numitul shell: DOS-Shell la Microsoft, Viewmax la Digital Research. Chiar dacă ele simplifică mult viaţa utilizatorului, nici una din ele nu pune realmente accente noi.

Un taskswitcher este de găsit atît în MS-DOS 5.0, unde e complet integrat în Shell, cît şi în DR-DOS 6.0, unde este un TSR numit Taskmax. Un taskswitcher (comutator de taskuri) nu are nici o legătură cu multitaskingul. El permite doar păstrarea simultană în memorie a mai multor programe şi comutarea dintr-unul într-altul. În timp ce unul lucrează, toate celelalte sînt "îngheţate". Taskmax permite în plus şi schimbul de date între programe. Cu ajutorul mouse-ului se selectează datele dintr-o aplicaţie, se comută şi apoi se copiază în cealaltă. Pentru ca datele să fie corect interpretate, Taskmax oferă 3 formate.

O găselniţă ce va bucura pe mulţi este Superstor, pe care Digital Research l-a cumpărat de la firma Addstor, integrîndu-l în furnitura DR-DOS 6.0. Ca şi alte programe de acest gen (de exemplu Stacker), Superstor promite dublarea capacităţii harddisk-ului, interpunîndu-se între controller şi DOS pentru a scrie/citi datele pe disc totdeauna în formă comprimată. Timpul care se pierde prin comprimare este recuperat din numărul micşorat de accese la disc, astfel încît, din acest punct de vedere, diferenţele sînt greu de sesizat. Programul "sstor.exe" divide harddisk-ul în două părţi, din care una rămîne să găzduiască fişierele a căror comprimare nu se doreşte. Restul sînt comprimate,iar la primul DIR utilizatorul se poate bucura descoperind o capacitate mărită a harddisk-ului. Dar nici Superstor nu face minuni: datele deja comprimate de "pkzip" sau "lharc" sînt stocate aşa cum sînt, nu mai pot fi comprimate. Şi fişierele cu grafică ale jocurilor, de regulă, nu mai pot fi comprimate. Cine are multe astfel de fişiere, va fi neplăcut surprins de cît de repede scade capacitatea afişată iniţial.

Securitate mărită a datelor oferă ambele sisteme de operare. Microsoft a cumpărat "mirror", "unformat" şi "undelete" de la Central Point Software (PC Tools), în timp ce Digital Research oferă "diskmap", "delwatch", "unformat" şi "undelete". "Mirror" şi "diskmap" salvează FAT-ul harddiskului, pentru a putea regăsi fişiere şterse din greşeală. Cît timp fişierele şterse nu au fost suprascrise, ele pot fi refăcute. "Delwatch" face chiar un pas mai departe, "simulînd" ştergerea înspre sistemul de operare - dar datele rămîn fizic disponibile. De abia cînd s-a umplut discul sau a fost atinsă limita specificată, ştergerea devine ireversibilă. În ambele cazuri, "unformat" face "nefăcută" o formatare. Spre deosebire de MS-DOS, DR-DOS oferă un "supliment" în ce priveşte securitatea şi protecţia datelor. Atît PC-ul cît şi fişiere individuale pot fi

protejate prin parole. Comanda "login" permite numai persoanelor autorizate accesul la calculator. "Password" atribuie fişierelor sau directoarelor cîte un cuvînt -cod şi unul din trei niveluri de acces. În plus, astfel utilizatorul îşi defineşte şi parola globală, utilizată în mod automat la accedarea fişierelor protejate. O afacere destul de complicată, care riscă să pună un utilizator distrat în situaţia de a-şi pierde toate datele.

Protecţia din DR-DOS nu este de netrecut (chiar dacă acest lucru nu mai e posibil prin simplă boot-are de pe o dischetă MS-DOS), dar oferă o protecţie suficientă pentru "nespecialişti".

"Diskopt" este un alt utilitar inclus în DR-DOS 6.0 şi care relevă grija Digital Research ca sistemul să cuprindă tot ce trebuie pentru o utilizare eficientă a PC-ului "Diskopt" încearcă să optimizeze accesele la harddisk, refăcînd în porţiuni continue fişierele fragmentate.

"Filelink" permite o legătură între două calculatoare folosind interfaţa serială şi un cablu standard.

Utilitarele "diskcopy" şi "diskcompare", în DR-DOS 6.0, folosesc în mod automat memoria extinsă şi memoria expandată sau un fişier ajutător pe harddisk. În fiecare din cazuri, discheta trebuie schimbată o singură dată - în MS-DOS, la un disc HD de 3,5" sînt necesare 7 schimbări de dischetă!. În DR-DOS, o dischetă-sursă poate fi copiată imediat de mai multe ori.

Ca editor, DR-DOS oferă un model ce foloseşte sintaxa Wordstar. MS-DOS-ul e mai atractiv la acest punct, utilizînd Quick-Basic-Editor.

O surpriză neplăcută cu "format a:". în DR-DOS 6.0: dischete noi-nouţe, dar cu porţiuni defecte fizic, sînt formatate fără nici un fel de mesaj de eroare. Duşul rece vine atunci cînd dischetele urmează să fie folosite... MS-DOS cade în extrema calaltă: cum dă de un sector defect, blochează toată pista. De aceea, pentru formatări, mai degrabă se folosesc alte programe (NU, PC Tools) decît cele incluse în furnituri.

În ce priveşte help-ul online, ambele sisteme au făcut progrese: toate utilitarele şi comenzile îşi dezvăluie scopul şi modul de utilizare dacă liniei de comandă i se adaugă parametrul "/?". MS-DOS oferă acest serviciu şi prin intermediul comenzii "help <nume -comandă>"; DR-DOS are un help interactiv numit "dosbook", care printr-un sistem de meniuri şi referinţe încrucişate permite găsirea informaţiei dorite. Digital Research a extins comenzile batch: "switch" permite preluarea unei taste de la utilizator, pentru a sări apoi la o etichetă specificată. Şi "config.sys"-ul poate fi (aproape) programat: sînt permise comenzi switch, salturi şi etichete. Astfel, PC-ul poate întreba la pornire ce configuraţie să folosească.

Digital Research furnizează un manual de utilizare de circa 700 de pagini, completat cu un volum mai mic cu indicaţii de folosire pentru Viewmax. Microsoft livrează un manual de utilizare de 800 de pagini, plus o scurtă introducere. Ambele manuale îşi îndeplinesc misiunea, oferind suficient ajutor mai ales la deciziile care privesc administrarea de memorie.

Mai rămîne problema compatibilităţii, pe care cel ce face testul o poate rezolva în două feluri: fie instalează toate softurile de care dispune, apasă cîteva taste, mişcă un pic mouse-ul şi încearcă să tragă nişte concluzii, (din păcate, incompatibilităţile de regulă nu sînt vizibile decît în anumite cazuri speciale, greu de verificat), fie caută de la bun început un soft "problematic" (caz în care riscul de a "nu merge" este la fel de mare sub MS-DOS 5.0 ca şi sub DR-DOS 6.0. Windows nu are nici un fel de probleme sub DR-DOS (cu condiţia ca înaintea instalării să se dea comanda "memmax -L -U"). Toate softurile încercate merg fără probleme. Ar mai fi de zis că între timp DR-DOS este aşa de răspîndit încît producătorii de soft se văd obligaţi să-şi testeze noile produse sub ambele sisteme.

În concluzie, care ar fi DOS-ul mai bun? Digital Research oferă, fără nici un dubiu, tehnici mai rafinate şi funcţii mai multe. MS-DOS face o impresie mai compactă, deoarece toate utilitarele folosesc aceeaşi suprafaţă SAA cu mouse şi meniuri pull-down. Dacă Microsoft se bizuie pe vechi cumpărători angrosişti, Digital Research caută să obţină contracte OEM (Original Equipment Manufacturer) şi ofertanţi de hardware ieftin.

Dar o trecere de la MS-DOS 5.0 la DR-DOS 6.0 (sau invers) probabil că nu merită (pentru cine trebuie să o plătească...). Cine porneşte acum, este probabil mai cîştigat cumpărînd DR-DOS 6.0. Accente realmente noi nu sînt puse nici de Microsoft, nici de Digital Research. De fapt, au fost integrate în furnitura de livrare utilitare pe care producători independenţi le oferă de mult. În ultimă instanţă, sînt vindecate numai simptomele care fac ca DOS să fie "trecut" şi inadecvat exploziei informaţionale din ultimii ani: un "taskswitcher" înlocuieşte un multitasking real, un optimizor de harddisk mai evită un pic slăbiciunile sistemului de fişiere bazat pe FAT şi gîndit de fapt pentru dischete, driver-ele de memorie creează iluzia utilizării lineare a spaţiului de adrese.

(I.F.)

Fabrica de zvonuri

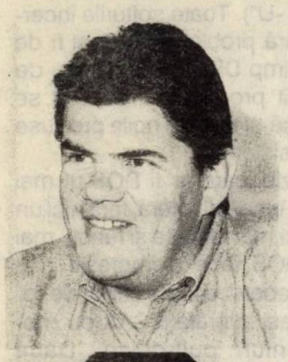
Prima reacţie la apariţia DR-DOS 6.0

Microsoft-ul dezvăluie cîteva din "secretele" viitoarei versiuni. Patru sînt inovaţiile decisive care urmează să propulseze Microsoft-ul în fruntea plutonului în cursa tehnologiilor: flat memory-model, noul sistem de fişiere, multitasking şi funcţii de reţea. Dacă asta vă aminteşte de OS/2, nu puteţi greşi. Într-adevăr, DOS 6.0 urmează să şteargă graniţele între DOS, Windows şi OS/2. Cel mai evident devine acest lucru în modul de administrare a memoriei: DOS foloseşte excesiv structura de segmente "prescrisă" de primele procesoare Intel. De aceea, memoria este divizată în bucăţele mărunte, astfel încît programatorul trebuie să jongleze nu numai cu registrele de adresă, ci şi cu alt segment. Cel tîrziu cu procesoarele 386 acest lucru a devenit necesar, dar, pînă acum, DOS era un sistem de operare axat pe 8088 în faţa căruia toate procesoarele erau egale. DOS 6.0 urmează să schimbe această stare de fapt. Noua lozincă este "jos divizarea memoriei, jos bariera celor 640 kocteţi". Sigur, este o întrebare dacă se va păstra compatibilitatea. Cum este sigur că numai posesorii de 386, 486 sau 586 vor putea să se bucure de noua administrare a memoriei. Tot ei, privilegiaţii, vor primi ca "supliment" un multitasking adevărat sub DOS. Nimic din toate acestea nu vor putea bucura posesorii de calculatoare cu 8088 sau 80286.

Nu vor exista însă diferenţe în sistemul de gestiune al fişierelor. Noul DOS va şti să lucreze cu diverse sisteme de fişiere, printre ele HPFS din OS/2. Un lucru pare cert de pe acum: sistemul de fişiere actual poate fi îmbunătăţit în aşa fel încît să păstreze "aspectul" DOS, dar să lucreze de două ori mai rapid.

Ca termen de apariţie pentru DOS 6.0 este planificat sfîrşitul anului 1992, chiar dacă acest lucru este la fel de nesigur ca şi lista definitivă a caracteristicilor şi numele "bebeluşului". Încă este posibilă redenumirea sistemului de operare (MS-DOS/2 e o propunere), integrarea în Windows 3.1 (sau 3.2 sau 4.0) sau abandonarea lui completă, datorită unor probleme tehnice insurmontabile.

Le roi est mort! Vive le roi!



Asthon Tate: Firma americană de software care a proiectat S.G.B.D-ul dBase, cel mai utilizat program de gestiune a bazelor de date. Această definiţie putea fi citită în 1984 într-un dicţionar de informatică. În schimb între algebra booleană şi Btx nu se găsea nici o definiţie despre Borland şi nici despre Kahn Philipp.

Probabil că această situaţie l-a supărat oarecum pe Philipp, născut în 1952, la Paris, într-o familie evreiasco-germană, în timp ce plonja în bazinul său de înot cu vedere spre plaja californiană. Probabil că şeful firmei Borland, destul de reţinut în ceea ce priveşte informaţiile despre viaţa sa personală - ştim doar că nu doarme decît patru ore pe noapte, că urăşte televiziunea, (nu ştim dacă pe motiv de emanueli), că nu se uită niciodată la televizor şi ca urmare cîştigă o mulţime de timp, că este un reputat muzician de jazz şi că este vegetarian convins - tocmai îşi consuma legumele pasteurizate la bordul propriului yacht de croazieră Dolphin Doncer. Probabil că nu dorea decît să se răzbune pe o instituţie americană, ca Ashton Tate de pildă, pentru faptul că Biroul American de Imigrări l-a declarat în 1985 emigrant ilegal.

Oricare ar fi fost motivul, la 10 iulie '91, Philipp Kahn a înghiţit firma Ashton Tate. Un peşte mare într-adevăr, plăcerea de a face cumpărături costîndu-l 440 milioane dolari. Dar de unde a avut atîţia bani, se întreabă cei din branşă? Ashton Tate a avut în anul trecut o cifră de afaceri de 230 milioane (pierderi: 18 milioane), Borland a avut o cifră de afaceri de 227 de milioane (cîştig: 27 milioane). Să fi fost firma IBM cea care i-a împrumutat sumiţa, doar pentru a înfige o sulă în coasta firmei Microsoft? Asemănător modului în care în 1983, Philipp înfiinţa în Scotts Valley firma Borland International, cu bani tapaţi de la prieteni?

Omului i s-a mai încredinţat o sarcină ingrată: cine nu-şi aminteşte de acţiunea juridică deschisă de Ashton Tate împotriva firmei Fox Software, care a eşuat deoarece un judecător american a declarat dBase-ul, din cauza viciilor de formă strecurate în copyright, liber ca păsările cerului? Judecătorul a făcut între timp recurs, din proprie iniţiativă, aşa că încă tot nu se ştie dacă vicleana vulpe, cu al ei program Foxbase (între timp Foxpro 2.0), mai bun şi mai rapid, a şterpelit sau nu ceva de la dBase.

Între timp s-a stabilit că firma Ashton Tate a produs singură produsul dBase, de la prima pînă la ultima instrucţiune, şi că nu a dezvoltat doar produsul Public - Domain elaborat de firma JPL pentru ministerul apărării.

Pînă la urmă nici Philipp Kahn nu se poate reţine de a mai arunca cîte o privire şi prin curţile vecinilor. Programul său de calcul tabelar de succes Quattro era destul de înrudit cu un alt program de calcul tabelar, care stăpînea piaţa şi care se cheamă Lotus 1-2-3.

Se făceau deja speculaţii, avocaţii îşi frecau bucuroşi mîinile, numai că la 11 octombrie '91 Departamentul Justiţiei (DOJ), din SUA, aprobînd achiziţionarea firmei Ashton Tate de către firma Borland International, a pus şi o condiţie: Borland să nu se mai judece cu Fox Software Inc. pentru drepturile de copyright pentru familia de produse dBase. DOJ afirmă că drepturile de copyright pentru structura meniurilor şi limbajele de comandă "promovează monopolurile în detrimentul inovaţiei".

Cînd valurile se vor linişti, Borland va deţine un plus de 50% pe piaţa sistemelor de gestiune a bazelor de date. Adunînd segmentele de piaţă deţinute în 1990 în SUA de Paradox şi de dBase se ajunge la un procent de 74.3%. La vînzări noi în '90, după IDC, situaţia prezentîndu-se în modul următor: 64% dBase, 17% Paradox, 11% Superbase, 4% Super DB, 3% Dataease, 1% Foxpro. Avînd în vedere faptul că în lume există peste 3 milioane de utilizatori dBase, în viitorul apropiat cele două produse vor coexista, aşteptîndu-se versiunile Windows atît pentru dBase cît şi pentru Paradox. Va exista un produs de migrare între cele două produse, care se va chema "Object dBase", denumit intern la Borland şi Turbo-XBase, produs care va fi compilator şi interpretor dBase în acelaşi timp, va fi compatibil dBase III şi dBase IV şi va avea o interfaţă Paradox.

Produsele Borland vor fi astfel concepute, după cum afirmă Philipp, încît operarea unui calculator să fie mai uşoară decît a unui casetofon. El afirmă că în viitor calculatoarele vor exista pentru a lucra aşa cum lucrează oamenii şi nu pentru a-i constrînge pe aceştia să lucreze aşa cum lucrează ele.

Dar să vedem cum a receptat vestea tranzacţiei Eduard Altmann, redactor de peste 10 ani la revista economică DM din Germania:

»"Firma Ashton Tate cumpărată de firma Borland", pe cine n-a şocat în urmă cu cîteva săptămîni această ştire? Ei bine, pe mine nu. Mărturisesc că sînt şi eu vinovat de această tranzacţie. Cel puţin moral. În mine am ştiut

că aşa se va întîmpla încă de mult, de cînd testînd pachetul încă fierbinte al noului, pe atunci, dBase 4.0, şi încercînd să sortez 300 de articole, am primit mesajul: "Not enough memory". Şi aceasta cu toate că am eliminat din memorie tot ceea ce se putea. Mi-am spus atunci: această firmă este în declin, trebuie să existe o pedeapsă.

Dar au fost şi momente frumoase cu dBase. Acestea au fost încă şi mai înaintate, cînd am primit o copie de dBase II aproape fără greşeli. În sfîrşit datele de pe harddisk-ul meu n-au mai fost trimise în nirvana de o putere necunoscută.

Îmi amintesc cu plăcere şi de dBase III+. Deoarece datorită programului "Assistent" n-am mai fost neajutorat şi am putut să-mi scriu programele linie cu linie, învăţînd să programez. Alţii au trebuit să plătească mulţi bani pentru aceasta, în timp ce eu n-a trebuit decît să-mi sacrific o parte din timpul liber. Oricum ce altceva aş fi avut de făcut serile şi la sfîrşit de săptămînă?

După cîţiva ani de studiu zadarnic a manualelor, mi-a pus cineva la inimă, mulţumescu-ţi ţie Philipp, un SGBD exotic denumit Paradox. La început neîncrederea a fost mare: o suprafaţă utilizator cu totul nouă, fişiere incompatibile şi ceea ce era mai rău un limbaj de programare străin, numit PAL. Numai din masochism profesional am încercat cu Paradox cîteva calcule statistice şi spre surprinderea mea în cîteva minute am obţinut, din pustiul de date, rezultate pentru a căror obţinere ar fi trebuit să lucrez la un program în dBase III zile întregi.

Începînd de atunci mi-a spus şi "Borland ceva. Chiar dacă generatorul de programe a rămas pentru mine o carte ferecată cu şapte lacăte, nedepăşit în filozofia sa secretă decît cel mult de PAL, şi încă n-am învăţat să-i iubesc suprafaţa utilizator antică cu comenzi îmbricate totuşi Paradox îşi face treaba fără blocaje şi fără programare.

În bucătăria de software a lui Philipp sfîrîie din nou ceva. Primele mirosuri ale unei noi mîncări numită Parabase răzbat înafară şi produc apetit: posibilitatea de a lucra în reţea, tempoul Paradox-ului, flexibilitatea lui Object Vision, confortul în operare al lui Windows şi utilizatorii lui dBase - miroase a inserare între algebra booleană şi Btx«.

(Romulus Maier)

De ce a cumpărat Novell Digital Research?

"Fuziunea este un pas absolut logic, căci produsele se completează" - declară un reprezentant Novell. Din fuziunea celor doi producători de sisteme de operare, care pînă acum vizau pieţe de desfacere distincte, se naşte un adversar serios pentru IBM, Microsoft, SCO, AT&T şi Apple. Căci, dintr-o singură lovitură, Novell devine un ofertant de sisteme de operare atît pentru locuri de muncă individuale (staţiile client în reţea), cît şi pentru server-e. Un rol decisiv joacă în acest cadru DR-DOS, apărut în versiunea 6.0. Netware se orientează după DOS, astfel încît pînă acum Novell depindea mult de Microsoft. Prin intermediul DR, Novell posedă acum o tehnologie DOS proprie şi poate juca un rol mai activ. Cu alte cuvinte: DR-DOS urmează să fie extins spre a fi completarea ideală pentru Netware.

Deoarece liniile de produse ale celor două întreprinderi se completează în mare măsură, se naşte întrebarea unor sinergii posibile (puterea comună să fie mai mare decît suma puterilor individuale). În cel puţin 6 domenii, cele două întreprinderi pot profita de experienţa celeilalte.

Front-end grafic: suprafaţa utilizator din Netware se bazează în continuare pe DOS. Un front-end grafic, care să uşureze munca utilizatorilor şi administratorilor de sistem, lipseşte. Sisteme precum Windows conlucrează cu Netware, dar nu sprijină automat funcţiile Netware. Ar fi posibilă o suprafaţă utilizator comună pentru DOS şi Netware, bazată pe GEM.

Funcţii de reţea în DOS: protecţie prin parole, blocarea de fişiere şi articole (file and record locking) sînt cuvinte străine unui programator DOS obişnuit, motiv pentru care, de regulă, pentru utilizarea în reţea trebuiesc programate versiuni speciale. Cu integrarea unor astfel de funcţii în DOS (şi astfel şi în Windows) administrarea reţelelor ar putea să se simplifice, pentru că toate programele DOS ar fi utilizabile şi în reţea.

Multitasking cu Netware: DR are multă experienţă în sisteme multitasking, iar în prezent sînt puţine case de soft care oferă NLM-uri (Netware Loadable Modules), cu ajutorul cărora programe independente să poată fi executate pe server. Poate că în viitor va exista o cale mai simplă pentru a scrie aplicaţii client-server pentru Netware.

Reţele ieftine (low-end-LANS): Novell şi-a manifestat deja dorinţa de a nu "abandona" piaţa reţelelor ieftine unor firme ca Artisoft (Lantastic). Cu Multiuser DOS, Digital Research posedă deja o soluţie în această direcţie.

Sisteme de operare în timp real: cu FlexOS, DR oferă un sistem de operare în timp real care se potriveşte de minune pentru supravegherea stocurilor, facturări şi urmărirea producţiei. O piaţă lucrativă pentru Netware.

Comenzi identice pentru DOS şi Netware: ceea ce la Novell se cheamă NDIR, NCOPY şi SALVAGE, se numeşte XDIR, XCOPY şi UNDELETE în DR-DOS. Cît timp încă?

Conform declaraţiilor făcute de reprezentanţi ai celor două firme, planuri exacte încă nu s-au stabilit. Dar este sigur că nu va trece mult timp pînă să apară primele fructe ale acestei fuziuni.

(I.F.)

Balanţe contabile

Legea obligă toate întreprinderile, indiferent de forma lor de organizare, să ţină o contabilitate în care să se regăsească toate tranzacţiile efectuate. Şi cum spectrul Gărzii Financiare pluteşte deasupra tuturor întreprinderilor şi întreprinzătorilor, fie că este vorba de magnaţi ai industriei gogoşilor, de proprietari de birt, de producători numărul 1 în şi în, sau de directori de combinate, centrale, etc., atunci cînd aceşti şerifi ai spaţiului contabilicesc fac o descindere, pe masă trebuie să apară şi balanţa contabilă. Şi atunci cînd aceasta nu este curată, trebuie deschisă punga.

Există contabili şi probabil că vor mai exista, care nu au încredere decît în situaţiile făcute de mînă. Dar există, şi au existat contabili care au acceptat faptul că acestea s-ar putea face şi altfel. Aşa au apărut primele balanţe pe calculator, implementate mai întîi, cu mult entuziasm, pe aşa-numitele maşini de facturat şi contabilizat, apoi cu acelaşi entuziasm rămas neştirbit pe toată gama de calculatoare existentă în ţară, de la TDF, M18, mini şi pînă la Felix, entuziasm care a luat parcă o şi mai mare amploare odată cu intrarea în ţară a PC-urilor. Din acele vremuri a rămas celebra zicală: "Daţi-mi un Cray şi să vedeţi ce balanţă vă implementez!"

După datele pe care le avem există la această oră în ţară sute de variante de balanţe contabile doar pe PC-uri, posibil chiar mii. Este mai ceva decît la chinezi, la cules de orez în luna mai, un colectiv de programatori, o balanţă contabilă, ba chiar două. Numai că nu este suficient să dai viaţă unei aplicaţii, trebuie s-o întreţii, nu poţi spune s-a murdărit puradelu, nu-i nimic, facem altul. Ne imaginăm cu groază reacţia provocată de o cît de mică modificare de legislaţie: toată floarea cea vestită a programatorilor făcînd aceleaşi modificări în programele proprii, originale.

Sîntem de părere că înainte de a începe proiectarea unei aplicaţii este bine să aruncăm o privire în jur pentru a vedea dacă nu există deja ceva ce ne-ar putea satisface nevoile. Credem că şi la noi va trebui să se creeze o industrie şi o piaţă de software. Nu ne îndoim de faptul că nu peste mult timp vom putea cumpăra programele de care vom avea nevoie aşa cum astăzi cumpărăm o carte sau o bucată de salam.

Desigur există şi contraargumente: diversitatea întreprinderilor este mare şi a cerinţelor contabililor aşişderea, amintim doar că unii contabili îşi ţin contabilitatea cu ajutorul jurnalelor contabile, în timp ce alţii folosesc fişele şah, nu toate documentele sînt tipizate şi nu-i uşor să provoci o modificare de tabieturi. Din experienţa trecutului ştim că nu de puţine ori întreprinderile care au preluat "de gata" anumite aplicaţii au trebuit pînă la urmă să le întreţină cu forţe proprii, uneori trebuind să depună mai mult efort decît dacă ar fi proiectat singure aplicaţiile cu pricina. Tot din experienţă mai ştim că încercările de realizare a unor fluxuri informatice, nu de puţine ori în prezenţa unor fluxurilor informaţionale deficiente, au fost încununate de succes foarte rar, reuşita datorîndu-se în acele cazuri, în mod substanţial, contactului direct dintre proiectant şi utilizator. Şi totuşi sîntem prea puţini pentru a ne putea permite să facem acelaşi lucru şi încercînd să facem totul, nu realizăm pînă la urmă mare lucru.

Încercăm, prin acest articol, să lansăm o serie de teste comparative de produse soft autohtone. Pentru subiectul ales, balanţele contabile, am testat patru produse provenind din tot atîtea localităţi. Am dori să subliniem de la început faptul că toate cele patru produse sînt operaţionale, au fost implementate cel puţin o dată şi fiind întreţinute în permanenţă, (aşa cum am fost asiguraţi), cu siguranţă se vor mai perfecţiona. Nu avem pretenţia de a vă prezenta cele mai bune patru produse existente la această oră în ţară şi nici de a face o prezentare exhaustivă a acestui gen de produse. (Este posibilă existenţa unor produse mai bune, iar o prezentare exhaustivă ar fi imposibilă).

Mărturisim că am încercat contactarea mai multor producători de astfel de aplicaţii şi că cei rămaşi în competiţie au cel puţin un merit, acela de a fi acceptat, la solicitarea noastră, focul întrecerii.

Aplicaţiile testate sînt foarte diferite atît în ceea ce priveşte aria de probleme acoperite cît şi ca modalitate de prezentare, limbajul de programare folosit etc. Cea mai simplă se rezumă la realizarea calculelor efectiv necesare în formularul de balanţă de verificare cu 8 coloane astfel:

- pornind de la solduri iniţiale sau total sume anterioare
- introducînd rulajele debitoare şi creditoare pentru fiecare cont
- editează situaţii tip balanţă cu total sume cumulate şi sold final calculat.

Extrema cealaltă caută să acopere pe cît se poate integral domeniul financiar contabil:

- se porneşte de la documentele de bază care sînt utilizate în domeniu
- se realizează evidenţa analitică la nivelul unor conturi
- se generează automat informaţii tip notă contabilă
- se asigură interfaţa către aplicaţii tip GESTOC şi MIFIX
- se editează rapoarte tip:
  - jurnale contabile
  - fişe tip şah
  - balanţe la nivel de conturi sintetice şi analitice
  - editare bilanţ contabil şi anexe la bilanţ.

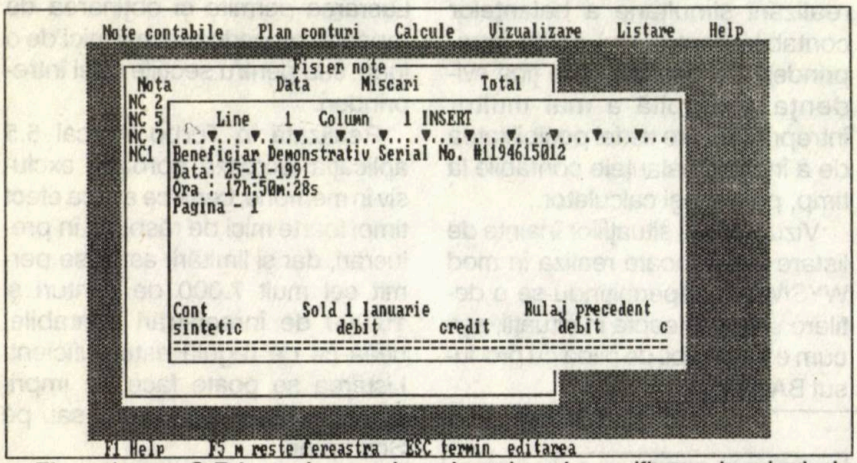
BALPRO

BALPRO este o balanţă contabilă elaborată de firma Lotus-S.R.L. din Brăila. Lucrarea este disponibilă în mai multe medii de programare: Turbo Pascal, C, Quattro, Foxbase, dBase III şi dBase IV.

Pornind de la solduri iniţiale, nomenclator (plan) de conturi şi note contabile, lucrarea permite vizualizarea şi/sau listarea balanţelor analitice şi sintetice, a fişelor şah, a notelor contabile încărcate şi permite, de asemenea, şi investigarea separată a operaţiilor din fiecare lună a anului.

Este singura lucrare testată care permite o configurare proprie a listelor şi care oferă utilizatorului, la nivel de help, şi informaţii privind reglementările în vigoare din domeniul contabilităţii şi un dicţionar de termeni permiţînd familiarizarea cu domeniul şi a utilizatorilor nespecialişti.

Aplicaţia permite, de asemenea, importul/exportul unor fişiere în format SDF şi dBase permiţînd preluarea datelor de balanţă şi în alte aplicaţii.

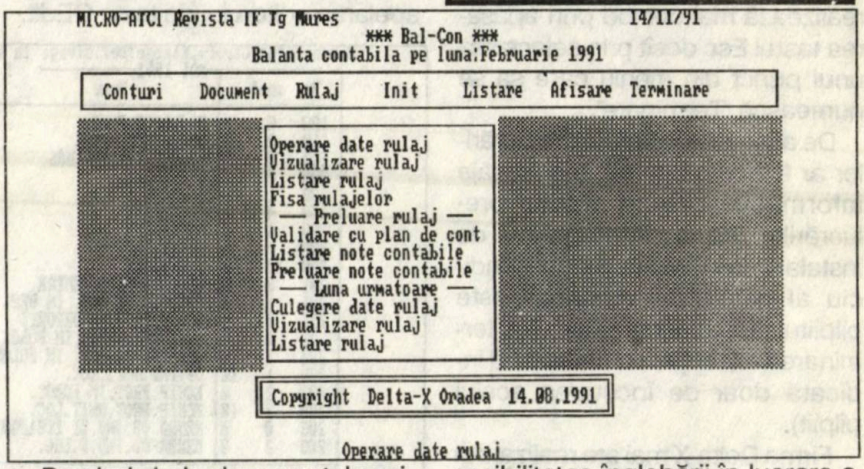


Firma Lotus S.R.L. mai pune la dispoziţia utilizatorilor şi aplicaţiile FIXPRO (mijloace fixe) şi SALPRO (salarizare/personal) şi are în faza de testare o aplicaţie care va integra gestiunea magaziilor, a recepţiilor de marfă, a vînzărilor către beneficiari, a facturărilor de marfă, urmărirea încasărilor, salarizare/personal, evidenţa mijloacelor fixe, balanţă contabilă. (Aplicaţia va fi disponibilă după 1.01.1992).

BAL-CON

Bal-Con este o aplicaţie destinată evidenţei contabile a unităţilor economice şi este elaborată de firma Delta-X SRL din Oradea.

Aplicaţia realizată cu ajutorul produsului Clipper face parte dintr-un complex de lucrări din care fac parte şi aplicaţiile MIFIX şi GESTOC.



Pornind de la documentele primare folosite pentru înregistrări contabile, note contabile, solduri iniţiale şi nomenclator de conturi, aplicaţia permite listarea balanţelor de verificare la nivel de conturi analitice şi/sau sintetice, a fişei sintetice şah, a extrasului de cont, a listei de venituri şi cheltuieli, a defalcării cheltuielilor pe capitole, etc.

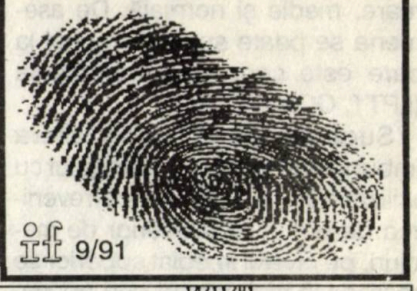
Lucrarea permite defalcarea conturilor sintetice pe patru nivele (max. 11 cifre) şi ca urmare la intrare pot fi folosite atît note contabile cît şi documente primare: carnete CEC, chitanţe, dispoziţii de încasare/plată, etc., ceea ce are ca efect posibilitatea înglobării în lucrare a unor evidenţe analitice la nivel de cont contabil, cum ar fi evidenţa mijloacelor băneşti, evidenţa debitori/creditori, etc. Se permite şi realizarea balanţelor pe secţii şi centralizarea acestora la nivel de întreprindere.

BAL-CON este singura aplicaţie testată care permite definirea unor corespondenţe de conturi şi efectuează validarea on-line în momentul introducerii datelor contabile.

Documentaţia produsului poate fi vizualizată în orice moment. Avînd programul Norton Guide instalat rezident, la o simplă tastare a combinaţiei de taste Shift-F1, se intră în documentaţie şi se poate naviga în ea pentru lămurirea problemelor dorite.

Din furnitura de livrare face parte şi un program de instalare care permite stabilirea parametrilor monitorului, a imprimantei, a parametrilor de implementare, iniţializare fişiere, indexare fişiere, refacere rulaj cumulat. Pot fi utilizate imprimante de tipul: Epson FX - 1050/850, RCD 9335 Scamp, EP-2001, NX-1500 STAR sau alte tipuri de imprimante, putîndu-se specifi-

ăl mai fain !



ca numărul de linii/pagină, numărul de coloane/linie şi secvenţele de control pentru densitate mare, medie şi normală. De asemena se poate specifica portul la care este conectată imprimanta (LPT1, COM1, COM2).

Sugerăm proiectanţilor cîteva îmbunătăţiri la nivelul dialogului cu utilizatorul: considerăm că revenirea de pe un nivel inferior de meniuri, pe nivelul imediat superior se realizează mai comod prin apăsarea tastei Esc decît prin selectarea unui punct din meniu care să se numească "Terminare".

De asemena în timpul prelucrărilor ar fi bine să se afişeze mesaje informative privind stadiul prelucrărilor (de ex. în programul de instalare, la indexare, singurul indiciu al acceptării comenzii este pîlpîitul LED-ului harddisk-ului, terminarea execuţiei comenzii fiind indicată doar de încetarea acelui pîlpîit).

Firma Delta-X mai are realizate şi aplicaţiile Mi-Fi - evidenţa şi calculul amortizării mijloacelor fixe şi GE-MA - gestiunea şi evidenţa contabilă a valorilor materiale, care împreună cu prezenta lucrare pot fi tratate ca un subsistem integrat. De asemenea în colaborare cu un colectiv de proiectare din Suceava este în curs de realizare o aplicaţie de personal-salarizare.

CONTAS

CONTAS - contabilitate analitică şi sintetică, este o aplicaţie elaborată de întreprinderea ELINCO SRL din Bucureşti, cu ajutorul produsului Foxpro.

Pornind de la nomenclatorul de conturi, solduri iniţiale, note contabile, facturi, dispoziţii de plată/încasare, se editează jurnalul înregistrărilor, balanţele maistru-şah debit/credit, fişa contului analitic/sintetic, balanţele de verificare analitice/sintetice. Aplicaţia are înglobate şi gestionarea facturilor şi a dispoziţiilor de plată/încasare. De asemenea în suprafaţa utilizator a programului este înglobată şi apelarea editorului de texte QEdit.

Cursor:↑↓→←,Ctrl→;Sterge;Del;sterge la stinga:BS;Inserare:Ins;sus/jos:PgUp,PgDn

MAI 1991 — 1

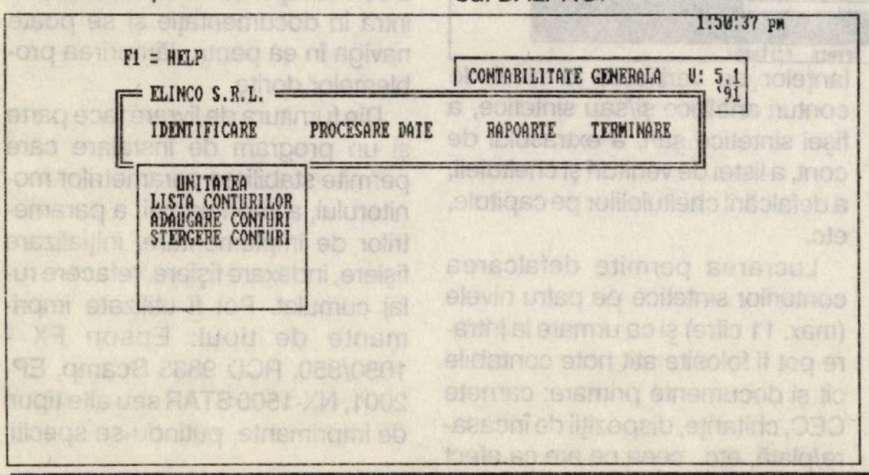
Nr. cont			Nume	Debit	Credit
100	0	0	MIJLOACE FIXE	823294.00	0.00
110	0	0	UZURA MIJL.FIXE	0.00	150420.00
123	0	0	INVESTITII IN CURS	823294.00	823294.00
200	0	0	MATERIALE	1592553.69	1713586.32
200	1	0	MATERIALE	295488.00	1291978.80
200	4	0	MATERIALE	0.00	0.00
200	6	0	MATERIALE	0.00	0.00
200	7	0	MATERIALE	0.00	0.00
200	19	0	MATERIALE	0.00	0.00
201	0	0	OBIECTE DE INVENTAR	152275.52	243215.26
201	1	0	OBIECTE DE INV. IN DEP.	152275.52	235563.86
201	2	0	ECHIP.PROT.IN DEPOZIT	0.00	7651.40
202	0	0	OBIECTE DE INV.IN FOLOS.	603236.37	183160.47
202	1	0	OBIECTE DE INV. IN FOLOS.	217463.55	183160.47
202	1	10	OB.INV.UNIT.LOC.	0.00	0.00
202	2	0	ECHIP.PROT.IN FOLOS.	385766.82	0.00
202	2	10	ECHIP.PROT.UNIT.LOC.	0.00	0.00
203	0	0	UZURA OB.INV.SI ECHIPAM.	153418.67	-351547.80
203	2	0	UZURA OB.INV.F.LOC.	0.00	0.00

F1-ajutor F2-Sortare↕salvare F3-Sterge F4-control F5-Salt la... F10-Exit

Editarea balanţei este condiţionată de existenţa egalităţii balanţiere.

O altă caracteristică importantă a acestui produs este posibilitatea realizării simultane a balanţelor contabile pentru mai multe întreprinderi. Un contabil care ţine evidenţa contabilă a mai multor întreprinderi are astfel posibilitatea de a încheia balanţele contabile la timp, pe acelaşi calculator.

Vizualizarea situaţiilor înainte de listare nu se poate realiza în mod WYSIWYG, nepermiţîndu-se o defilare stînga-dreapta în situaţii, aşa cum este posibil de pildă cu produsul BALPRO.



Produsul nu s-a comercializat pînă acum separat, fiind oferit gratuit întreprinderilor care au cumpărat calculatoare de la firma Elinco SRL.

BALL

BALL - balanţă expert, este o aplicaţie realizată de ing. Flaidăr Alin şi comercializată de întreprinderea Micro ATCI Tîrgu Mureş.

Avînd la intrare nomenclatorul de conturi, soldurile iniţiale şi notele contabile se permite listarea balanţelor sintetice/analitice, a situaţiei conturilor şi a fişelor şah. Lucrarea permite şi obţinerea de rapoarte pe perioade mai mici de o lună, sau pentru secţiile unei întreprinderi.

Realizată în Turbo Pascal 5.5 aplicaţia lucrează aproape exclusiv în memorie, ceea ce are ca efect timpi foarte mici de răspuns în prelucrări, dar şi limitări; astfel se permit cel mult 7.000 de conturi şi 10.000 de înregistrări contabile, ceea ce de regulă este suficient. Listarea se poate face pe imprimante compatibile Epson, sau pe Scamp-uri.

Este în curs de realizare o nouă versiune în Turbo Vision, care va efectua validarea conturilor introduse (ceea ce în versiunea actuală nu se realizează), va include tehnici de editare multidocument, listare în background cu PrintManager şi o structurare mai avansată a informaţiilor.

mat. Galfi Nicolae
ing. Maier Romulus

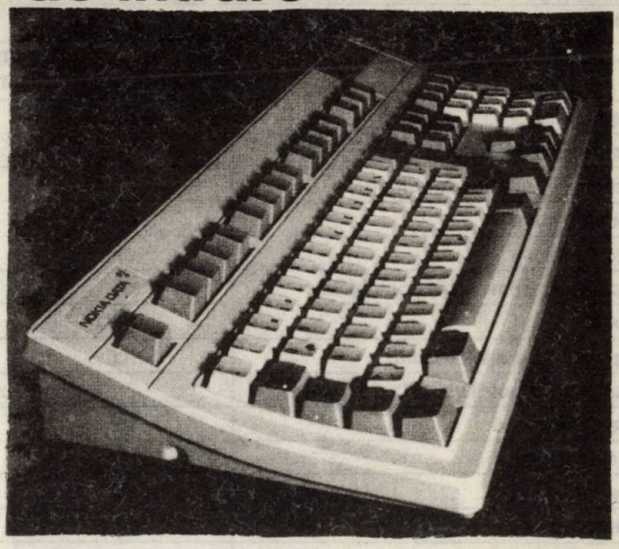
Balanţe contabile

Produs	BALPRO	BAL-CON	CONTAS	BALL
Producător	**Lotus S.R.L. Brăila** Tel. 946/35544	**Delta-X Oradea** Tel. 991/34257	**Elinco Bucureşti** Tel. 90/175300/175	**Micro ATCI Tg. Mureş** Tel. 954/31660
Preţ (lei)	20.000	55.000	-	30.000
Documente de intrare:				
Nomenclator (plan) de conturi	da	da	da	da
Solduri iniţiale	da	da	da	da
Note contabile	da	da	da	da
Documente primare din contabilitate	nu	da	nu	nu
Facturi	nu	da	da	nu
Dispoziţii de plată/încasare	nu	da	da	nu
Situaţii care pot fi vizualizate/listate*				
Balanţa analitică	da	da	da	da
Balanţa sintetică	da	da	da	da
Fişa şah	da	da	da	da
Note contabile	da	da	da	da
Extras de cont	nu	da	nu	da
Lista de venituri şi cheltuieli	nu	da	nu	nu
Defalcarea cheltuielilor pe capitole	nu	da	nu	nu
Situaţia conturilor	nu	nu	nu	da
Rulaje pe conturi	nu	nu	da	nu
Fişa contului	nu	nu	da	nu
Grafic anual	da	nu	nu	nu
Documentaţie/help-uri, interfaţă utilizator				
Reglementări în vigoare	da	nu	nu	nu
Dicţionar de termeni	da	nu	nu	nu
Îndrumar implementare/utilizare	da	da	da	da
Ferestre help generale	da	da	da	da
Help contextual	nu	parţial	nu	nu
Linie informativă (status)	da	da	da	da
Taste funcţionale	da	da	da	da
Calculator de buzunar	nu	nu	nu	nu
Cerinţe hardware (configuraţie minimă):				
Tip calculator	XT	XT	AT	XT
Memoria minimă necesară	nu/recomandat	nu/recomandat	da	nu/recomandat
Harddisk necesar ?	256 KB	512 KB	640 KB	256 KB
Spaţiu disc	cca. 300 KB	cca. 1,2 MB	cca. 1,2 MB	cca. 240 KB
Alte caracteristici:				
Număr maxim conturi	nelimitat	nelimitat	nelimitat	7.000
Număr maxim operaţii contabile	nelimitat	nelimitat	nelimitat	10.000
Utilizabil în reţea	nu	nu	da	nu
Interfeţe cu alte aplicaţii				
Mijloace fixe	da (SDF/dBase)	da	nu	nu
Gestiunea stocurilor	nu	da	nu	nu
Facturare	nu	nu	da	nu
Salarii	da (SDF/dBase)	în lucru	nu	nu
Altele				
Protecţie	certificare	certificare	dischetă cheie , parolă	certificare
Mediu de programare	Turbo Pascal, C Quattro, Foxbase dBase III, dBase IV	Clipper	Foxpro	Turbo Pascal

* Notă: Unele dintre situaţiile enumerate sînt asemănătoare principial, avînd doar nume diferite

Periferice de intrare

Articolul care urmează va prezenta perifericele de intrare care stau astăzi la dispoziţia utilizatorului, modul lor de funcţionare, care este utilitatea lor practică şi care va fi viitorul acestor interfeţe dintre om şi maşină. Unele dintre perifericele care urmează au fost tratate mai detaliat în numere anterioare ale revistei "if", de altele ne vom mai ocupa în numerele viitoare. Ceea ce ne interesează în momentul de faţă este o privire de ansamblu asupra întregului.



Tastatura - cel mai utilizat periferic de intrare

Chiar şi unul dintre cele mai bune calculatoare, un 80486/33 MHz cu procesor cache şi cu alte accesorii care îi măresc performanţele poate fi doar atît de bun, cît îi permite soft-ul şi modul în care este deservit. De multe ori, datele provenite de la alte calculatoare pot fi transferate automat, dar fără o introducere manuală a datelor un PC nu poate fi folosit nici în stadiul de astăzi al tehnicii.

Din păcate omul este foarte lent în comparaţie cu calculatorul, în plus el are nevoie de aparate (echipamente periferice) de intrare convenabile. De aceea se încearcă să se pună la dispoziţia utilizatorului unui calculator un aparat de intrare cît mai necomplicat şi uşor de manipulat. În acest sens este valabilă regula: cu cît sînt mai complexe datele de intrare ale unei anumite aplicaţii, cu atî trebuie să fie mai complex aparatul de intrare. Deci pentru intrări simple este nevoie de un aparat de intrare simplu.

De-a lungul ultimilor ani s-a dezvoltat combinaţia dintre mouse şi tastatură. Ambele aparate de intrare combină, printr-un compromis, propriile lor avantaje şi dezavantaje. Nu în ultimul rînd la acest lucru a contribuit şi adaptarea softwareului. Introducerea datelor via tastatură şi mouse este sprijinită de Windows 3.0 de pe calculatoarele MS-DOS, de suprafeţele grafice de sub Unix, de exemplu Motif, Sunview sau Open Desktop cît şi de Apple Macintosh. Dar mai sînt şi alte aparate de intrare care ajută omul în introducerea rapidă, necomplicată şi simplă a datelor într-un calculator.

Pentru preluarea textelor tipărite se foloseşte un scanner cu soft OCR (Optical Character Recognition) corespunzător. Cu acesta imaginile sînt digitalizate şi prelucrate în calculator.

Un joc, un robot sau o simplă conducere (comandă) pot fi deservite foarte bine cu un joystick, iar pentru aplicaţiile grafice există tablete digitizoare şi mouse-uri. Pe lîngă acestea mai există în anumite domenii ecrane sensibile la atingere, creioane optice (lightpen) sau combinaţii ale tuturor mediilor de intrare. Există deja, deşi încă în faşă, şi introducerea vocală prin microfon.

Principiul de funcţionare al aparatelor uzuale de intrare, cine le utilizează şi cît de eficace este interfaţa om-maşină pentru utilizator vor fi arătate în continuare:

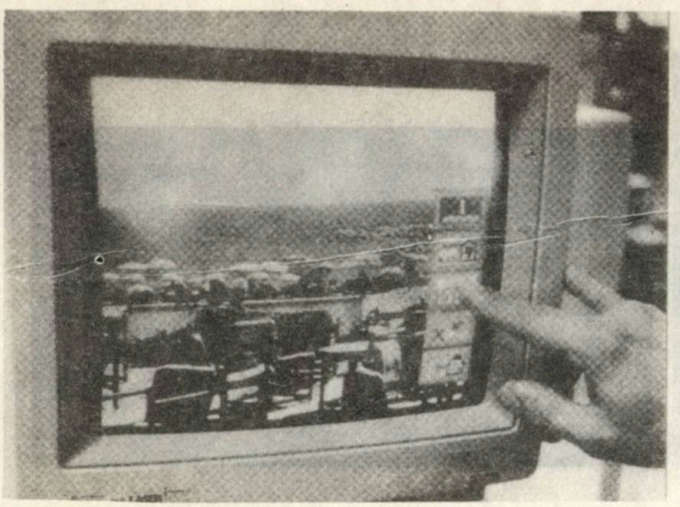
Tastatura
Ergonomia lasă de dorit

Nici un calculator nu este utilizabil fără tastatură. Tastatura este fără nici o discuţie cel mai des folosit aparat de intrare pentru calculatoare. Aceasta se explică prin faptul că pentru elaborarea textelor s-a folosit maşina de scris încă înaintea erei calculatoarelor. Era deci de la sine înţeles să se preia sistemul maşinii de scris. Din punct de vedere ergonomic o tastatură este extrem de neprietenoasă. Utilizatorul trebuie să-şi ţină mîinile într-o poziţie care nu permite o muncă relaxată.

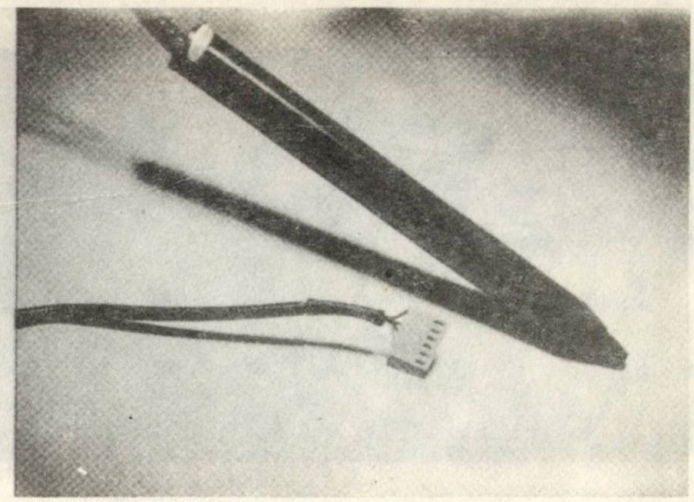
Deoarece arătătorul şi degetul mijlociu ale mîinii drepte (la dreptaci) sînt cele mai folosite, tastele cele mai des folosite ar trebui să se afle în dreapta acestora. De exemplu "J" nu face parte dintre acestea. Tastele cele mai des folosite se află în partea stîngă a rîndului superior al tastaturii.

Încercările de a obţine o tastatură cu adevărat ergonomică au fost respinse de piaţă, fiind adoptată doar poziţia tastaturii faţă de mîinile omeneşti, neţinîndu-se cont de frecvenţa de apariţie a fiecărei taste în parte. Şi pe viitor tastatura va rămîne aparatul de introducere numărul 1. Pentru a nu încărca calculatorul cu administrarea tastaturii, aceasta este dotată cu un procesor propriu. Acesta transformă semnalele electrice datorate apăsării unei taste într-un semnal serial pe care îl transmite prin cablul tastaturii calculatorului.

Calculatorul este dotat cu o interfaţă serială pentru tastatură;



Touchscreen: introducere prin apăsarea unor cîmpuri



Lightpen -în vîrful său se află o fotodiodă

aceasta este interogată de sistemul de întreruperi iar informaţia tastaturii se transmite spre BIOS. Momentan există două tipuri, tehnic diferite, de tastaturi. Una este pentru PC-urile 8086 iar cealaltă pentru AT-urile 80286, 80386 şi 80486. Aceste tastaturi nu sînt compatibile datorită diferenţelor hardware. Deci o tastatură PC nu funcţionează pe orice AT şi invers. Dar există şi tastaturi care posedă un comutator şi care pot funcţiona cu ambele tipuri de calculatoare. Deoarece la calculatoarele High-End, deci 386 şi 486 cu frecvenţa de tact ridicată, tactul de bus este mărit, se poate întîmpla ca procesorul tastaturii să nu poată alimenta destul de rapid interfaţa serială. Rezultatul: semne greşite sau lipsă de semne. Deci la cumpărarea unui calculator rapid trebuie găsită şi tastatura adecvată.

Cine-şi doreşte probleme de contact?

Pretenţiile relative la mecanica tastaturii sînt foarte ridicate. Ea trebuie să suporte multe milioane de tastări fără ca să apară defecţiuni mecanice.

Există 2 tipuri uzuale de contacte electrice. Mai ieftine sînt contactele cu folii, care au în componenţă 3 folii. Cea de jos are rolul de contact electric, cea din mijloc are rol de distanţare, iar cea de deasupra reprezintă cel de-al doilea contact electric. În locul de contact, folia mijlocie este prevăzută cu o gaură atît de mare încît folia superioară să nu o poată atinge pe cea inferioară decît la apariţia unei forţe de apăsare. Mai scumpe şi mai complicate sînt contactele cu lamele metalice. În acest caz apăsarea tastei apropie două lamele metalice şi realizează astfel contactul. În funcţie de modul de execuţie lamelele sînt suflate cu argint şi/sau aur, pentru a le prelungi perioada de viaţă şi pentru a micşora rezistenţa de contact. În cazul tastaturilor din industrie foarte des folosite în locul contactelor se folosesc şi cristale piezoelectrice. Acestea înregistrează tensiunile apărute la cele mai reduse presiuni şi le valorifică. Aceste cristale sînt dispuse sub o placă de oţel, astfel încît tastaturile Heavy-Duty pot fi construite în stil antiexploziv şi antiacvatic.

Există o problemă hard şi soft şi anume "blocarea" tastelor. Dacă o tastă este apăsată prea tare, ea poate sări înapoi şi să facă în acel moment un al doilea contact. Aproape fiecare tastatură are altă bătaie a tastelor, datorită modului diferit al execuţiei mecanice. Importantă este definirea unui punct de presiune; unele tastaturi scot un sunet la apăsarea fiecărei taste care poate fi deranjant. Nu există nici o tastaură perfectă, şi fiecare om are propria sa imagine despre cum ar trebui să fie o tastatură bună, de aceea înainte de cumpărare este bine să scrieţi, de probă, un text mai lung şi în final să lăsaţi să decidă degetele.

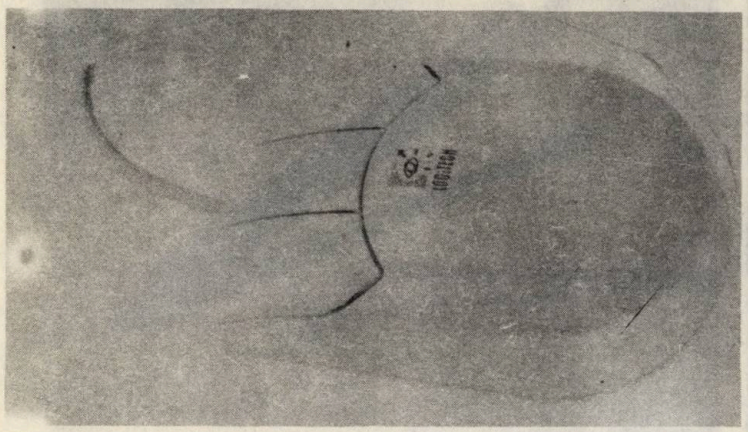
Ecranul - introducere directă

Deoarece ochiul este îndreptat spre ecran este de la sine înţeles să se facă "intrări" direct pe ecran. Cele mai importante două sisteme de intrare sînt ecranul sensibil la atingere (Touchscreen) şi creionul optic (Lightpen). Ambele sisteme permit utilizatorului, prin simpla apăsare pe ecran, direct sau cu ajutorul lightpen-ului, să lanseze programe. Bineînţeles că softwareul trebuie să fie adaptat acestui lucru.

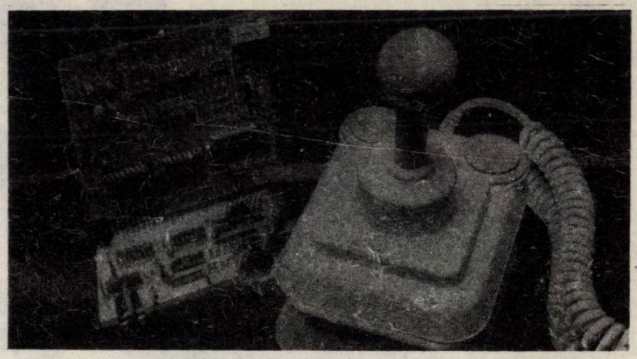
De regulă aceste sisteme sînt oferite azi la echipamentele şi terminalele publice (de exemplu pe aeroporturi, în birouri de turism şi în bănci), deoarece acest sistem economiseşte o tastatură şi utilizatorul nu trebuie să execute de regulă intrări complicate. Astfel se oferă unele puncte de meniu, dintre care se poate face alegerea prin apăsare directă.

Cu degetele obosite nu se poate scrie

Nici Touchscreen şi nici Lightpen nu pot fi folosite în mod util pentru intrări exacte şi de mai lungă durată; deoarece ecranul este aşezat vertical în faţa utilizatorului, mîna trebuie să fie liberă şi să nu se sprijine pe nimic. Acest lucru duce la o obosire rapidă.



Mouse - un periferic de intrare indispensabil



Joystick-ul nu este gîndit doar pentru jocuri

Singura soluţie ar fi integrarea ecranului în masa de scris. În cazul ecranului sensibil nu se pot obţine introduceri foarte exacte datorită diferitelor presiuni exercitate de degetele omeneşti. Aici intrarea se reduce la atingerea de cîmpuri mari. Pentru a face un ecran sensibil la tastare ei va fi acoperit cu mai multe straturi de folii. Aceste folii sînt formate din mai multe contacte extrem de subţiri, care sînt legate cu conductoare tot atît de subţiri. La atingere rezistenţa de contact a unui anumit domeniu se modifică, şi astfel pot fi calculate coordonatele X şi Y.

Lightpen-ul foloseşte el însuşi semnalul de pe ecran pentru a-şi determina poziţia. În vîrful lightpen-ului este aşezată o fotodiodă, care preia semnalul luminos avînd o frecvenţă de 50 Hz (sau chiar mai mult, în funcţie de frecvenţa de repetiţie a imaginii). Printr-o sincronizare exactă cu apariţia în timp a semnalului de pe ecran, poziţia lightpen-ului poate fi determinată foarte exact. În acest fel se obţine, prin alegerea unei fotodiode extrem de mici, o rezoluţie comparabilă cu rezoluţia ecranului. Şi în viitor touchscreen-ul şi lightpen-ul vor fi folosite doar rar, în aplicaţii speciale.

Modul de utilizare pare în primul moment foarte plăcut, dar lucrul îndelungat este obositor şi inexact. De aceea mouse-ul balistic va îndepărta aceste aparate de intrare orientate pe ecran.

Mouse-ul: sens şi nonsens

Mouse-ul, (v. "if" 3/90), îi împarte pe utilizatori în două tabere. Mulţi utilizatori care cunosc PC-urile încă de la începuturi cunosc anumite programe, cum ar fi Wordstar, încă de la primele versiuni. Au învăţat pe dinafară combinaţiile de taste şi pot lucra fără să mai ia mîinile de pe tastatură. Şi astăzi multe editoare de texte încă mai folosesc convenţiile Wordstar. Ele au fost într-o anumită măsură un standard. Mouse-ul ar trebui să uşureze munca, nemaitrebuind memorate combinaţii lungi de taste. Dacă a reuşit să fie prietenos, rămîne de văzut. Mouse-ul are sens doar dacă mîna nu trebuie să se mişte mereu între mouse şi tastatură.

Dacă dorim o salvare intermediară rapidă în timpul unei prelucrări de texte, în Word 5.0 trebuie să acţionăm combinaţia de taste (Ctrl - F10). În timpul în care se apasă aceste taste un alt utilizator apasă o singură dată o tastă a mouse-ului. Pe de altă parte este de la sine înţeles că un software specializat pe utilizare prin mouse este foarte prietenos. Nu degeaba suprafaţa utilizator Windows 3.0, a lui Microsoft, are un asemenea succes.

Mouse-ul este aparatul de intrare ideal în cazul prelucrării datelor grafice , deci CAD sau DTP. Între timp au apărut cele mai diverse forme de mouse. Baza se pune pe modul în care mouse-ul se adaptează mîinii. Dar producătorii uită de multe ori un lucru: există un număr destul de mare de stîngaci, ale căror nevoi ar trebui luate în seamă.

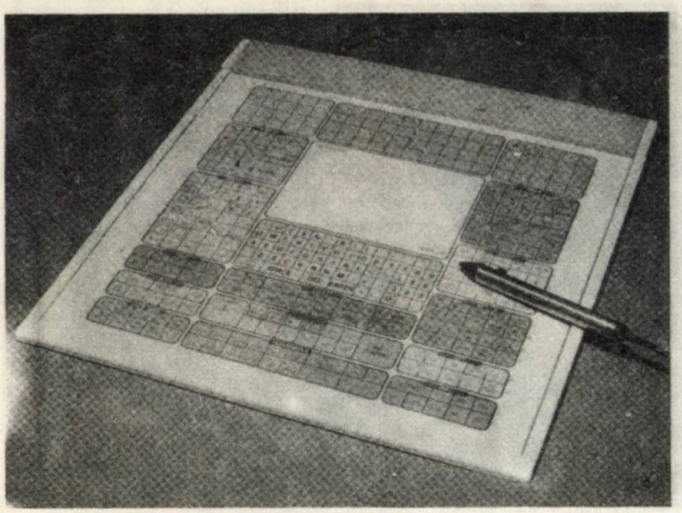
Casa şoricelului

Tehnica unui mouse este simplă. O sferă de cauciuc aflată pe partea inferioară a mouse-ului antrenează nişte rotiţe în direcţiile X şi Y. Mai există şi mouse-uri la care aceste două rotiţe sînt antrenate direct, deci fără sfera de cauciuc. Pe rotiţe sînt fixate cîte o plăcuţă găurită. O unitate formată dintr-o fotocelulă şi un LED transformă rotaţia în impulsuri electrice.

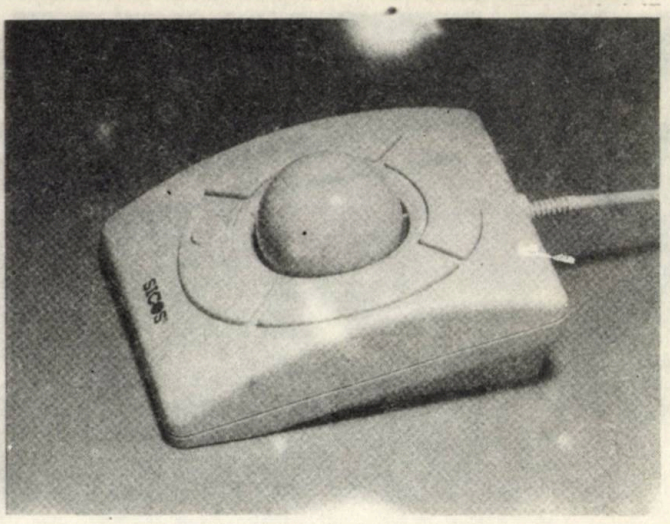
Transmiterea impulsurilor se face de regulă printr-un cablu care este conectat la interfaţa serială a calculatorului. Alţi producători de mouse economisesc cablu şi trimit impulsurile prin lumină infraroşie sau prin unde calculatorului. Cablul nu mai incomodează, în acest caz, mişcarea mouse-ului, în schimb mouse-ul devine mai greu datorită electronicii mai complicate şi a propriei alimentări cu curent prin baterie.

Mai există şi aşa-numitul bus-mouse. Acesta nu-şi trimite datele prin interfaţa serială, ci direct în bus-ul de date al calculatorului.

Primele mouse-uri au avut o rezoluţie slabă şi o adresare liniară.



Tableta grafică - un periferic de intrare foarte precis



Trackball - un mouse răsturnat

Mouse-urile moderne au o rezoluţie de pînă la 1200 dpi (Dots per Inch) şi se adresează balistic. Aceasta înseamnă că mişcînd uşor mouse-ul cursorul poate fi plasat pe ecran exact. La o deplasare rapidă a mouse-ului cursorul se va deplasa mult pe ecran, se economiseşte timp şi se foloseşte o suprafaţă mică pe mousepad (suprafaţa, pe care se mişcă mouse-ul).

Deoarece tot mai multe soft-uri se orientează pe operarea cu mouse, acestea devin din ce în ce mai utilizate. Rămîne de sperat, ca deservirea prin tastatură, care de multe ori este mai rapidă şi mai confortabilă, să nu aibă de suferit din acest motiv.

Joystick-ul, nu doar pentru jucători

Joystick-ul, (v. "if" 5/91), a avut mare succes atunci cînd piaţa HC-urilor s-a extins foarte mult. El se pretează foarte bine la introducerea valorilor de comandă uni şi bidimensionale. Multe jocuri au nevoie de un joystick. La îndrăgirea lui a condus şi preţul. Un joystick simplu poate fi găsit la mai puţin de 30 DM.

Dar joystick-ul îşi are locul nu doar în domeniul jocurilor, ci şi în comanda proceselor şi la deservirea maşinilor moderne. Acolo unde pe timpuri o macara era deservită cu nişte manete mari de comandă, astăzi este folosit joystick-ul, deoarece comanda se face acum electronic.

Există două tipuri diferite de joystick-uri. Cel mai simplu, folosit în domeniul jocurilor, este "joystick-ul digital". Maneta joystick-ului este încastrată articulat şi are pe fiecare din cele 4 părţi din interiorul carcasei cîte un întrerupător cu contacte care se închid şi se deschid în funcţie de mişcarea manetei. Contactele pot fi realizate cu lamele metalice ieftine sau cu microîntrerupătoare scumpe. Bineînţeles că nici unui joystick nu-i lipseşte un microîntrerupător suplimentar, denumit şi tastă declanşatoare (trăgaci).

Cel de-al doilea tip constructiv este "joystick-ul analog". Contactele simple sînt înlocuite cu două potenţiometre dispuse perpenticular. Astfel se pot introduce semnale proporţionale. Aceste joystick-uri se folosesc la comanda de la distanţă a modelelor. Semnalele de la joystick sînt transmise calculatorului prin intermediul unui gameport, deci a unei interfeţe pentru jocuri. Deoarece jocurile pe calculator sînt foarte îndrăgite, joystick-urile vor rămîne şi în continuare cele mai folosite medii de introducere.

Digitizorul: o concurenţă pentru mouse

Nu numai datorită faptului că ele ajung să-şi merite tot mai mult preţul, digitizoarele, (v. "if" 3/91), îşi fac tot mai mult apariţia pe piaţă. Pentru aplicaţii CAD aceste aparate de intrare sînt ca şi predestinate. O tabletă grafică constă dintr-o suprafaţă plană din material sintetic sub care se află un tastator electronic şi dintr-un creion special pentru introducerea şi comanda punctelor din meniu.

Fiecare poziţie a aparatului de intrare de pe tableta grafică se determină, în cele mai multe cazuri, inductiv sau capacitiv. Creionul emite unde electromagnetice, care sînt recepţionate de tabletă. Un convertor A/D digitalizează această informaţie şi o transmite mai departe calculatorului.

Unii producători oferă un protocol de transfer compatibil cu mouse-ul Microsoft, care dă posibilitatea utilizării lor şi în cazul unor aplicaţii care recunosc doar mouse. Deoarece cele mai multe tablete lucrează prin metoda inductivă, ele produc un uşor cîmp magnetic. Dischetele care se ating cu acestea pot fi deci afectate. Alte tablete grafice lucrează cu ultrasunete. Creionul emite ultrasunete care sînt înregistrate de două microfoane. Pe baza diferenţei de timp poate fi determinată poziţia exactă a creionului.

Pentru o mai simplă utilizare a tabletelor grafice se livrează odată cu ele suporturi adaptate pentru aplicaţii speciale, cum ar fi AutoCAD. Astfel se pune la dispoziţia

Touchmouse - pentru introducere sînt necesare obiecte ascuţite

Scanner-ul -poate prelua atît texte cît şi imagini

utilizatorului un mod de comandă care nu funcţionează prin meniuri ecran ci prin atingerea unor simboluri de pe tabletă. Se pune întrebarea dacă acest mod de comandă are sens deoarece continua mutare a privirii de pe ecran pe tabletă şi înapoi devine în timp foarte obositoare. Pentru posesorii de PC-uri Notebook sau a altor calculatoare portabile şi pentru cei care nu au loc pentru un mouse sau o tabletă grafică pe biroul lor, se oferă alte două alternative de aparate de intrare: în primul rînd este vorba de Trackball, în principiu un mouse culcat pe spate şi în al doilea rînd de aşa-numitul Touchmouse, o tabletă grafică miniaturizată.

Pentru economisirea spaţiului: Touchmouse şi Trackball

Astfel utilizatorului i se dă posibilitatea deservirii unor programe, care necesită o conducere prin mouse, şi fără suprafaţa plană.

Trackball constă dintr-o sferă înglobată într-o carcasă deschisă în partea superioară. La fel ca şi la mouse mişcările sferei în direcţiile x şi y sînt preluate de două rotiţe. Se produce un semnal serial înregistrat cu ajutorul unei fante luminoase şi a unei rotiţe găurite, care va fi pus la dispoziţia programului utilizator. Deservirea unui trackball este mai greoaie decît a unui mouse. Mişcările mîinii nu pot fi traduse direct, ceea ce înseamnă că mîna trebuie să corecteze mereu, şi deci vor exista erori de poziţionare. De aceea introducerea datelor cu un trackball se foloseşte doar în aplicaţii speciale, cum ar fi comanda maşinilor şi a sistemelor CAD speciale. Relativ nou pe piaţă este touchmouse-ul, o tabletă mică, şi utilă, unde intrarea se face cu unghia sau cu alte obiecte ascuţite. Contactele integrate permit determinarea poziţiei locului apăsat, şi software-ul de comandă calculează cu aceste date poziţia mouse-ului pe ecran. Touchmouse-ul nu se pretează pentru o poziţionare exactă, este însă o alternativă bună pentru cei care vor să utilizeze Windows 3.0 pe un laptop şi care nu doresc să ducă cu ei şi un mouse. Bineînţeles că trackball şi touchmouse nu sînt singurele aparate de intrare care fac concurenţă mouse-ului. Unii producători au proiectat între timp o mulţime de înlocuitori. În schimb nu prea este de aşteptat ca utilizatorii să le accepte cu uşurinţă, deoarece, de regulă, omul este supus inerţiei şi este refractar la nou.

Convertorul A/D: o punte spre electronică

Fără îndoială că cea mai bună metodă de comunicaţie ar fi "calea directă" de la lumea exterioară către calculator. Dacă calculatorul ar înţelege vocea, ar putea prelua semnale de măsură şi ar putea percepe prin intermediul unei camere de luat vederi mediul înconjurător, multă muncă de introducere s-ar putea economisi. Aceste idei nu mai sînt de domeniul SF, căci cercetările în această direcţie sînt foarte avansate, iar în cazuri speciale există deja soluţii acceptabile. Un senzor oarecare, fie el microfon, cameră de luat vederi sau indicator de temperatură, înregistrează un semnal. Dar toate semnalele de origine naturală sînt analogice, valorile lor nu se modifică în trepte ci în mod continuu. Un sunet nu este deci puternic sau încet, ci el poate avea o infinitate de nuanţe, şi în acest caz calculatorul are probleme. El înţelege de fapt doar valorile digitale, deci "cu curent" şi "fără curent". Dacă un calculator urmează să prelucreze mărimi analogice, acest lucru presupune o digitalizare prealabilă. Pentru aceasta se foloseşte aşa-numitul convertor A/D; acesta împarte un anumit domeniu de tensiune într-un număr de intervale. Fiecărui interval i se atribuie apoi în calculator o anumită valoare. Cu cît este mai mare numărul de intervale cu atît pot fi convertite mai precis valorile semnalului de intrare. Cele mai des întîlnite procedee de transformare a semnalelor analogice în semnale digitale sînt:

- procedeul integrării duble (Dual-Slope)
- procedeul aproximării succesive şi

- procedeul conversiei fulger (Flash)

Principiul conversiei Dual-Slope constă în măsurarea timpului necesar unui condensator pentru a se încărca la tensiunea pe care trebuie s-o convertească. Cu cît tensiunea de intrare este mai mare cu atît mai lung este timpul de încărcare necesar. Acest procedeu este relativ lent şi se pretează doar pentru măsurarea unei tensiuni cu un voltmetru digital, dar nu şi pentru digitalizarea vocii sau a imaginii. În acest caz procedeul aproximării succesive este mult mai rapid; în acest caz, în principiu, se activează o tensiune de verificare în cadrul convertorului A/D, care se adaptează tensiunii efective şi apoi are loc conversia. Această conversie se pretează deja pentru digitalizarea vocii.

Multimedia: un cuvînt magic

Semnalele video pot fi digitalizate prin procedeul flash. În acest caz tensiunea de intrare este aplicată în acelaşi timp la mai multe comparatoare de tensiune, al căror număr depinde de lăţimea eşantionului. Viteza de conversie a acestui sistem este de circa 20 de nanosecunde. Prin comutări electronice rapide şi exacte s-a ajuns, din punct de vedere tehnic, la procedeuri de înregistrare digitală a sunetului pentru discuri CD şi pentru discuri optice. Şi cuvîntul magic "Multimedia" este în ultimul timp foarte des folosit. Cu multimedia se pot prelucra în calculator în timp real imagini video, acest lucru fiind imposibil fără un convertor A/D rapid. Şi în tehnica măsurărilor convertoarele A/D ocupă un loc important. Acolo unde înainte trebuiau introduse manual lungi şiruri de măsurători pentru a le valorifica prin calculator, astăzi se leagă un senzor la calculator prin intermediul unui convertor A/D, iar calculatorul preia atît colectarea cît şi prelucrarea datelor.

Tehnica de producţie şi de automatizare nu mai pot fi imaginate fără astfel de sisteme. Căci tehnicile flexibile de automatizare se pot realiza doar prin controlul exact al desfăşurării evenimentelor. Chiar şi digitalizarea imaginii şi a vocii se pot face doar printr-un convertor A/D. Cu toate că din punct de vedere tehnic s-a ajuns ca aceste informaţii să poată fi digitalizate prelucrarea lor, cum ar fi: recunoaşterea vocii sau recunoaşterea vizuală a unui obiect cu ajutorul calculatorului sînt încă în faşă. Deci va trebui să mai treacă încă puţin timp pînă cînd vom putea dicta un text calculatorului.

PC-ul ca cititor pasionat

Punerea problemei este simplă: un text scris pe hîrtie trebuie "introdus" într-un calculator, pentru a fi prelucrat în continuare. Există două soluţii posibile. Prima ar fi copierea textului, dar această soluţie necesită mult timp. A doua ar fi citirea unui astfel de text şi aducerea lui cu un software adecvat în format text. Acest lucru îl face scanner-ul, el poate "introduce" atît imagini cît şi grafice în calculator. Aceste aparate de citire descompun o imagine sau un text în puncte singulare digitale şi trimit această informaţie PC-ului. Există mai multe metode de digitalizare a unei imagini. Cele mai importante sînt: procedeele Multiplier, Tehnic, Laser şi CCD. Cel mai utilizat procedeu este CCD cu matrice linie şi matrice bidimensională. Suprafaţa de scanat se vizualizează pe întreaga lăţime a scaner-ului cu un tub luminiscent. Lumina reflectată va fi preluată apoi de o serie de fotodiode şi diode laser. Ele înregistrează diferenţele luminos - întunecos şi le convertesc într-o combinaţie binară. În funcţie de densitatea elementelor CCD rezoluţia scannerului este dinainte fixată. Scannerele normale pun la dispoziţie o rezoluţie de 300 dpi (Dots per Inch). La fel cum există mai multe procedee există şi mai multe tipuri de scannere, cele de format mare fiind folosite mai ales în domeniul profesional iar cele de mînă în cel particular. În comparaţie cu tehnica avansată a scanner-elor recunoaşterea scrisului mai poate fi încă îmbunătăţită. O imagine scanată este memorată în prima fază în format grafic de calculator. Pentru prelucrarea ulterioară într-un editor de texte ea trebuie transformată mai întîi într-un format text. Acest lucru se face cu un software OCR (Optical Character Recognition) care recunoaşte optic semnele, (v. "if" 2/90). Tocmai în cazul recunoaşterii scrisului se vede diferenţa dintre om şi calculator. Un calculator, cu programele sale cu tot, nu poate să facă asociaţii. Astfel programele OCR simple nu pot recunoaşte decît litere foarte clare. Literele care prezintă greşeli datorită unor spoturi calitativ inferioare îi dau multă bătaie de cap programului de recunoaştere. Dar şi texte scanate foarte curat necesită programe bune pentru a executa transformări neeronate. O metodă rapidă şi sigură este aşa-numita "Pattern Matching". Pentru fiecare literă în parte se memorează unul sau mai multe "modele normale". Un nou semn va fi apoi suprapus peste model şi comparat prin înlănţuire logică. În funcţie de cum au fost date toleranţele de comparaţie, se poate întîmpla, ca o literă de fapt clară să nu fie recunoscută de program sau un C să devină un E.

Tehnica prin suprapunere s-a dovedit foarte bună în cazul documentelor tipărite şi scanate foarte bine, avînd o viteză mare de prelucrare şi erori puţine. Pentru documentele scrise de mînă sau cursiv procedeul Feature-Recognition este mai bun. Acesta se orientează după caracteristicile unei litere. Caracteristica tipică C este că nu conţine nici o linie dreaptă şi că spre dreapta este deschis. Algoritmul de urmărire a conturului conţine caracteristicile tipice ale literei pe care le valorifică. Din păcate acest procedeu este foarte lent şi nu merită folosit la textele tipărite.

(I.M.)

Touch me, user!

Noi şi foarte variate alternative la tastaturile clasice au stîrnit în ultima vreme interesul utilizatorilor de PC-uri. Ceea ce în industrie a devenit deja ceva normal, a ajuns să fie oferit şi utilizatorilor particulari din domeniul PC-urilor. Cuvîntul magic al viitorului este: atingere.

În curînd în fiecare birou de turism: Informaţii via Touchscreen

Atingerile nu sînt omise

"Touchpanels" este titulatura generală pentru acest mediu de introducere şi este denumirea pentru o suprafaţă (din material sintetic, sticlă, folie) care prin atingere cu mîna, degetul sau un creion poate fi folosită pentru comandarea funcţiunilor programului. Denumirea de Touchpanel vine din limba engleză şi s-ar traduce prin cîmp de atingere sau cîmp de senzori (touch = a atinge, panel = cîmp, tablă de instrumente, tablă de comutaţie).

Dintre domeniile în care s-a folosit pînă acum putem aminti medicina şi în primul rînd industria. În tot mai mare măsură această tehnică este folosită atît pentru schimbul general cît şi pentru cel special de informaţii. Pe aeroporturi călătorii sînt informaţi prin Touchscreen despre rutele de zbor timpii de decolare şi de aterizare şi despre alte lucruri specifice, prin simpla atingere a unui cîmp din meniul de pe ecran, care dă posibilitatea prin submeniuri la o apropiere tot mai mare de informaţia dorită.

Nici birourile de turism nu-şi vor mai informa clienţii în viitorul apropiat cu ajutorul cataloagelor color despre ofertele de concediu, ci tot prin "Touchscreen" vor oferi meniuri pentru concedii, care vor cuprinde imagini de hoteluri, ştranduri şi camere de hotel. Pe lîngă domeniul turistic vor mai fi dotate cu tehnica de atingere multe suprafeţe utilizator din domeniul PC-urilor. Dar nu doar ecranele pot fi echipate cu această tehnică ci şi tastaturile pot fi echipate cu senzori sensibili la atingere, caz în care tastele se înlocuiesc cu senzori, care sesizează o atingere foarte uşoară cu degetul.

Un semn cu degetul ajunge

Datorită tehnicii de touchscreen utilizatorul poate interveni pe instrumentul de introducere date. Monitorul îi deschide o nouă dimensiune a "introducerii interactive". Dar cum ajunge ecranul mat al monitorului să devină mediu de introducere?

Particularităţile suprafeţei monitorului nu rezolvă problema. Componentele care răspund de aceasta sînt un ecran, care se află pe aceasta, sau deja implementat în monitor şi un controller. Ecranul sensibil la atingere (sticlă, material sintetic, folie) este prevăzut cu un strat transparent şi bun conducător de electricitate introdus prin difuzie (un praf metalic foarte fin), iar pe conturul ecranului sînt prevăzuţi electrozi care sînt montaţi între ecran (tub cinescopic, sau tip plasma) şi rama ecranului.

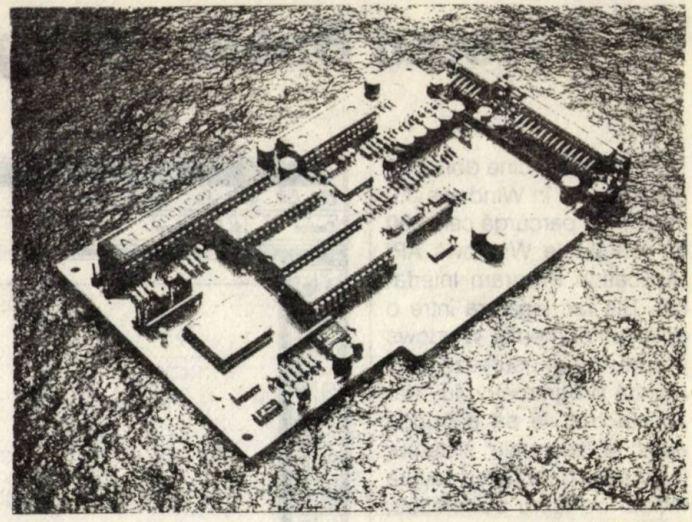
Această sticlă stratificată şi transparentă, cu conductori electrici, are electrozii amplasaţi liniar şi alimentaţi cu o tensiune alternativă. Între aceşti electrozi se află suprafaţa activă, sensibilă la atingere.

Prin atingerea suprafeţei active cu degetul ia naştere un curent de conducţie care se scurge spre pămînt prin intermediul corpului uman. Această tehnică este denumită "analog-capacitivă". Curentul are valori de doar cîţiva nA.

Rezoluţia senzorilor este de 1024 de puncte de atingere per axă, în interiorul cîmpului calibrat de pe ecran. Construcţia unei astfel de suprafeţe monitor, sensibilă la atingere, nu mai este o problemă. Există astfel de touchscreen pentru toate tipurile de monitoare, cu tuburi cinescopice, cu plasmă electroluminiscentă sau cu cristale lichide, cu suprafaţă bombată sau plată.

Inima propriu-zisă este aşa-numitul "touchcontroller", care este montat fie direct în carcasa moni-

torului, fie poate fi introdus ca o cartelă de extensie (de 8 biţi) în calculator. Controller-ul înregistrează atingerea ecranului, calculează poziţia în care acesta a fost atins şi transmite rezultatul, sub forma unor date seriale, calculatorului pentru a le prelucra mai departe. În cazul în care controller-ul este montat direct în carcasa monitorului comunicaţia cu calculatorul se realizează prin intermediul unei interfeţe seriale RS232.



Inima: fără Touchcontroller nu funcţionează nimic

Touchcontroller-ul constă din două procesoare, procesorul analog şi procesorul de comunicaţie. Partea analogă a touchcontrollerului este constituită din nivelul de intrare al senzorilor. Cei patru senzori sînt măsuraţi continuu ceea ce duce la generarea unui curent între ecran şi pămînt, la atingerea ecranului. Din curenţii de conducţie rezultaţi pentru fiecare electrod sînt calculate căderile de tensiune între electrod şi punctul de atingere, tensiuni care sînt preluate în continuare. Pentru ca semnalul să fie cît mai puternic se face o filtrare a frecvenţelor iar conductorii sînt ecranaţi împotriva încărcărilor electrostatice.

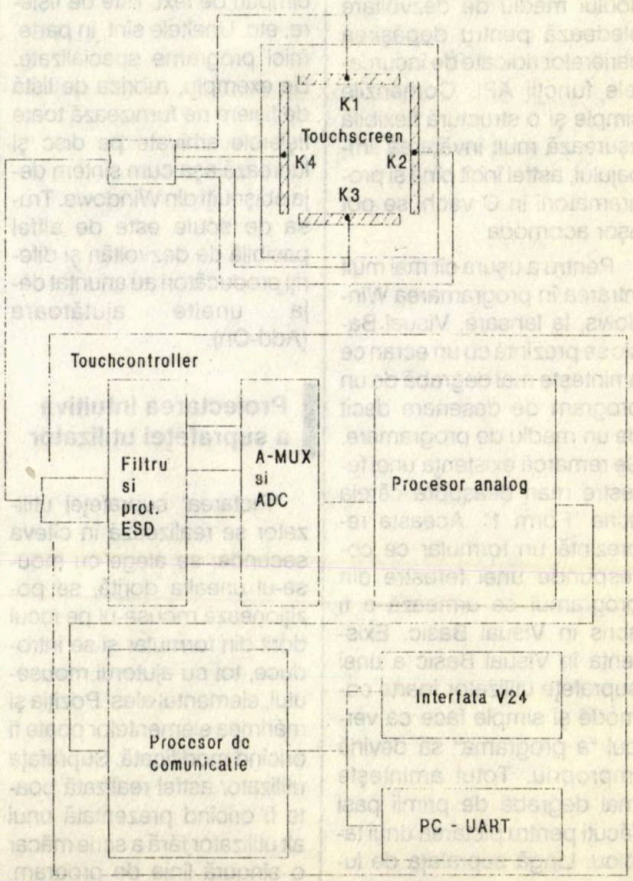
Procesorul senzorilor de atingere constă dintr-un convertor analog/digital, un interpretor de comenzi şi o interfaţă spre procesorul de comunicaţie. Semnalele de măsură analogice sînt digitalizate şi sînt transmise mai departe ca şi coordonate x, y. Acestora le corespund coordonatele punctului de atingere, motiv pentru care întreaga suprafaţă activă trebuie calibrată. Aceste date calibrate sînt memorate într-un EPROM şi sînt utilizate la calcularea coordonatelor.

Unitatea de comunicaţie controlează transmisia datelor prin interfaţă (fie interfaţa serială RS232 pe un bus de 8 biţi, fie interfaţa externă V24). Ea trebuie să supravegheze în permanenţă întregul controller şi să stabilească definitivarea mai multor cîmpuri de atingere.

Senzorii de atingere sînt plasaţi sub o suprafaţă care îi protejează la zgîrieturi şi care nu permite pătrunderea prafului sau a umezelii. Ca aparat de intrare în domeniile în care utilizatorii nu sînt specialişti în calculatoare, un "deget arătător" este un mijloc de comunicaţie indispensabil. Prin intermediul lui, practic oricine poate comunica cu calculatorul. Pentru ecranele sensibile la atingere au apărut între timp drivere speciale. Pentru utilizatorii din mediul MS-DOS (de exemplu pentru Windows 3.0) există deja drivere speciale cu ajutorul cărora utilizatorul îşi poate defini comenzile de intrare.

Tot ceea ce este posibil cu aceste ecrane sensibile la atingere va dovedi viitorul. Cu siguranţă utilizatorul va trebui să devină mai "sensibil", pînă într-o zi în care calculatorul îi va spune:
"Touch me, user" - dar te rog nu prea tare!

(R.M.)



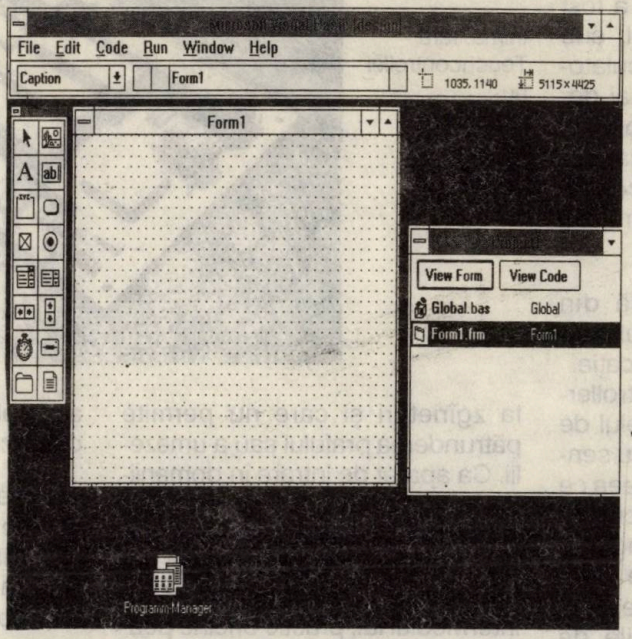
Reprezentarea schematică a unui Touchcontroller şi a unui Touchscreen. Touchcontroller-ul constă din două procesoare: procesorul analog şi procesorul de comunicaţie.

Visual Basic - programare intuitivă

Pînă acum, cine dorea să programeze în Windows 3.0 trebuia să parcurgă cele 600 de funcţii ale Windows API (Aplication Program Interface). Ele fac legătura între o aplicaţie şi mediul Windows propriu-zis şi uşurează mult munca programatorului, dar acesta trebuie să pătrundă adînc în "secretele" sistemului. Înainte de a putea savura modul de lucru cu ferestre, programatorul trebuia să înveţe modurile de apel şi sintaxa acestora, ceea ce necesita un consum de timp, efort pe care doar puţini programatori sînt dispuşi să-l depună.

Prin lansarea Visual Basic-ului, Microsoft speră ca aceste dificultăţi să aparţină definitiv trecutului, proiectarea unor aplicaţii Windows fiind mult uşurată. Deşi nu vor putea fi realizate aplicaţii de genul Excel, de exemplu, totuşi, întreaga concepţie a noului mediu de dezvoltare pledează pentru depăşirea barierelor ridicate de încurcatele funcţii API. Comenzile simple şi o structură flexibilă uşurează mult învăţarea limbajului, astfel încît pînă şi programatorii în C vechi se pot uşor acomoda.

Pentru a uşura cît mai mult intrarea în programarea Windows, la lansare, Visual Basic se prezintă cu un ecran ce aminteşte mai degrabă de un program de desenare decît de un mediu de programare. Se remarcă existenţa unei ferestre mari deasupra căreia scrie "Form 1". Aceasta reprezintă un formular ce corespunde unei ferestre din programul ce urmează a fi scris în Visual Basic. Existenţa în Visual Basic a unei suprafeţe utilizator foarte comode şi simple face ca verbul "a programa" să devină impropriu. Totul aminteşte mai degrabă de primii paşi făcuţi pentru pictarea unui tablou. Lîngă suprafaţa de lucru se găseşte "trusa de scule" (Toolbox) ce cuprinde toate elementele grafice cunoscute din Windows: butoane, cutii de control, cîmpuri de text, liste de fişiere, etc. Uneltele sînt, în parte, mici programe specializate. De exemplu, rubrica de listă de fişiere ne furnizează toate fişierele arhivate pe disc şi lucrează aşa cum sîntem deja obişnuiţi din Windows. Trusa de scule este de altfel pasibilă de dezvoltări şi diferiţi producători au anunţat deja unelte ajutătoare (Add-On).



Proiectarea intuitivă a suprafeţei utilizator

"Pictarea" suprafeţei utilizator se realizează în cîteva secunde: se alege cu mouse-ul unealta dorită, se poziţionează mouse-ul pe locul dorit din formular şi se introduce, tot cu ajutorul mouse-ului, elementul ales. Poziţia şi mărimea elementelor poate fi oricînd modificată. Suprafaţa utilizator astfel realizată poate fi oricînd prezentată unui alt utilizator fără a scrie măcar o singură linie de program. Modificările se pot realiza doar cu cîteva mişcări ale mouse-ului.

În Visual Basic elementele au denumirea de controale. Utilizatorul are libertate deplină în alegerea şi aranjarea obiectelor, astfel încît acelaşi tip de control poate fi pus în acelaşi formular de mai multe ori. De exemplu, în aceeaşi fereastră pot exista mai multe rubrici de dialog sau butoane. Fiecare element va fi tratat ca un obiect independent cu caracteristici proprii (mărime, lungime, culoare). Aceste caracteristici vor fi denumite "proprietăţi". Proprietăţile unui anumit element pot fi vizualizate sau definite cu ajutorul cutiei de proprietăţi. În afară de proprietăţi, fiecare obiect este definit printr-un nume pentru a putea fi apelat în mod univoc. Acest fapt este esenţial avînd în vedere că pe parcursul programului elementele definite (controalele) pot fi folosite ca şi simple variabile. E indicat ca elementele definite să fie botezate cu nume ce aminteşte de funcţia obiectelor (ex: "Pornirea ceasului"). În caz contrar Visual Basic-ul denumeşte el însuşi elementele cu nume tipice care, în cazul apariţiei multiple a aceluiaşi tip de control, vor fi numerotate crescător (ex: Button 1, Button 2, etc.). Pentru apelarea unui control din program este suficient să se dea numele controlului urmat de un punct, drept separator, şi urmat de o proprietate. În acest fel, în caz de nevoie, se poate modifica o proprietate a unui control (ex: mărimea sau culoarea).

Sînt necesare cunoştinţe de programe

Alfa şi omega programării Windows o constituie tratarea aşa-numitelor evenimente (events). Programele clasice funcţionează după o schemă clar definită. Nu se poate spune acelaşi lucru despre programele Windows. Acelaşi program Windows poate fi folosit în mod diferit de utilizator, în funcţie de diferitele aplicaţii. Înseamnă că programul trebuie să reacţioneze flexibil la acţiunea utilizatorului. Nucleul Windows sesizează în timpul rulării unui program dacă survine un anumit eveniment. Un astfel de eveniment îl poate constitui selectarea cu un clic pe mouse a unui buton de comandă sau acţionarea unei taste de către utilizator. Dacă survine un astfel de eveniment, el sensibilizează o rutină specifică evenimentului.

Visual Basic este altfel conceput. După ce s-a creat suprafaţa utilizator, va fi dat, pentru fiecare obiect, programul ce urmează a fi rulat la alegerea obiectului. La un dublu clic pe element se deschide o fereastră în care Visual Basic inserează de la început două instrucţiuni: începutul şi sfîrşitul subrutinei.

(continuare în pag. 38)

Ce este bioinformatica?

O abordare biologică a dezvoltării de programe

Dezvoltarea de programe cu ajutorul metodelor biologiei, iată domeniul bioinformaticii. În ultimii ani au fost dezvoltate mai multe astfel de tehnici şi tehnologii, unele dintre ele, deja demonstrate, au ieşit pe piaţă, altele se mai perfecţionează încă în laboratoare. Toate oferă o mare promisiune programatorilor: de a putea depăşi barierele actuale de complexitate şi dimensiune a programelor şi de a obţine soluţii simple pentru o mulţime de probleme spinoase, ca, de exemplu, recunoaşterea formelor. Cel mai cunoscut succes al acestor tehnici sînt reţelele neuronale cunoscute sub mai multe denumiri, ca procesare distribuită paralelă sau conecţionism, sau neuroinformatică. Alte tehnologii ale bioinformaticii sînt algoritmii genetici, sistemele "L", logica fuzzy, sistemele fractale, automatele celulare, dinamicile haotice. Încercăm, credem, pentru prima dată să considerăm aceste tehnici împreună. Pe măsură ce învăţăm mai multe despre ele, legăturile ne apar din ce în ce mai clare.

În unele cazuri, tehnicile sînt legate la nivel teoretic, de exemplu dinamicile haotice, fractalii, sistemele "L" şi sistemele funcţii iterate, au acelaşi suport matematic. În alte cazuri, tehnicile sînt distincte, dar păstrează totuşi legătura cu lumea naturală, biologică.

Nevoia de bioinformatică

În anii '80 sistemele de reţele în timp real, GUI-urile, (suprafeţele utilizator grafice), şi aplicaţiile distribuite au dominat piaţa, cu o complexitate şi mărime a programelor extrem de variate. Mulţi dintre noi am întîlnit în aceşti ani limitele metodelor actuale de dezvoltare a programelor.

De exemplu, cele zece milioane de linii de program scrise pentru naveta spaţială au ajuns la un blocaj catastrofal în timpul numărătorii inverse de la lansare, datorită lipsei unei virgule. În anul trecut, întreaga reţea de telefoane AT&T a fost imobilizată o zi întreagă datorită unei condiţii anormale întîlnită de către software-ul multiprocesorului.

Sau, mai aproape de noi, multe dintre firmele producătoare de software pentru PC-uri au fost stînjenite de programe care rulau prea încet sau care aveau erori (de exepmlu dBASE IV 1.0, LOTUS 1-2-3/G, OS/2, Windows, Macintosh System 7, etc.). Cu cît programele sînt mai mari, cu atît mai tare se blochează şi cu atît mai mult timp se pierde cu aceste blocaje.

Deci, nu pare surprinzător că specialiştii în informatică au căutat metode de construcţie a unor sisteme software mai puţin fragile, mai sigure, mai flexibile care să-şi conserve nivelul înalt de funcţionalitate. Au fost explorate multe direcţii, de la programarea orientată pe obiecte la metodele inteligenţei artificiale. Una dintre cele mai interesante abordări este bioinformatica.

Natura ştie cel mai bine

Să nu fim înţeleşi greşit, scopul bioinformaticii nu este să construiască programe mari. Cele mai multe implementări actuale sînt mici în dimensiuni, şi consumă puţine resurse. Dar cu cît numărul de linii ale unui program convenţional se măreşte, cu atît complexitatea lui se apropie de un bioorganism mic. Sistemele vii au structurile interioare construite din componente simple (celule) cu un grad mare de redundanţă, tolerante la greşeli şi foarte adaptabile. Biostructuri de o incredibilă complexitate sînt construite cu ajutorul unei mulţimi foarte reduse de reguli. Specificaţiile pentru o fiinţă umană de şase picioare înălţime (molecula genetică ADN) încap într-un container mai mic decît a milioana parte dintr-un club cu latura de un inch, şi totuşi fiinţele umane sînt mult mai complexe decît un Boeing 747. Mai mult, spre deosebire de naveta spaţială noi nu avem blocaje cînd o virgulă lipseşte din secvenţa ADN (de obicei).

Hans Moravec, un teoretician de calculatoare şi cercetător în robotică, a încercat să compare puterea de calcul a organismelor naturale, cu CPU-urile artificiale, după părerea lui un MIPS (milion de instrucţiuni pe secundă) este aproximativ echivalent cu 100.000 celule neuronale din creier. Altfel spus un Macintosh este echivalent cu un melc, iar un Cray2 este la nivelul unui rozător mic.

Putem extinde comparaţia hardware şi în domeniul software-ului. Lanţul ADN al unei fiinţe umane, este aproximativ de şase bilioane biţi lungime, şi reprezintă specificaţiile de construcţie ale unui om. Pare a fi foarte multă informaţie, dar de fapt este surprinzător de puţină dacă o comparăm cu produsele software contemporane. ADN-ul este o secvenţă de cod şi date doar de o mie de ori mai mare decît informaţia conţinută în produse de tipul dBASE IV sau Lotus 1-2-3. Cu siguranţă că programatorii naturii nu folosesc limbajul C şi nici asamblorul!

De fapt informaţia utilă într-un ADN este mult mai mică pentru că doar 5% din ADN este folosit ca date de către corpul uman. Aceasta înseamnă aproximativ 40 Mbytes, mărimea unui hard disc de PC-AT. Celelalte procente, ADN-ul vechi, conţin informaţii neutilizate de sistemul ţintă. O parte din aceste date pot fi văzute ca biţi de stop sau de paritate, folosiţi la corecţia erorilor şi protejarea semnalului genetic. Altă parte reprezintă informaţii depăşite, valabile în trecut pe scara evoluţiei fiinţei umane, cu alte cuvinte o informaţie demodată. Cu siguranţă, programatorii naturii ştiu să folosească ifdef-urile pentru

a comenta codul de care nu mai este nevoie.

Reţelele neuronale artificiale sînt bazate pe modul în care teoriile actuale descriu funcţionarea creierului natural, adică prin interconectări de celule neuronale. În realitate funcţionarea creierului mamiferelor continuă să rămînă necunoscută biologilor, dar acest fapt nu a stagnat dezvoltarea modelului de reţele neuronale.

Ariile de aplicabilitate ale reţelelor neuronale sînt surprinzător de extinse, în jurul principalului obiectiv de recunoaştere a formelor. Unul dintre primele modele de reţele neuronale, elementul linear adaptiv (ADALINE) al lui Bernard Widrow a fost folosit încă din 1959 ca filtru adaptiv hardware pentru eliminarea ecourilor pe liniile telefonice. În ultimii ani, reţelele neuronale artificiale au fost folosite pentru recunoaşterea formelor, procesarea imaginilor, analiza limbajului natural, filtrarea zgomotelor, controlul roboţilor şi modelarea financiară. Un avantaj cheie al reţelelor neuronale este modul în care acestea imită puterea creierului de a lua decizii şi de a trage concluzii cînd este confruntat cu informaţii complexe, cu zgomote, irelevante şi parţiale. Abilitatea reţelelor neuronale de a înţelege datele măsurabile ale lumii reale a devenit critică atunci cînd s-a încercat lucrul cu echipamente ţinute în mînă cum ar fi prelucrarea scrisului de mînă sau a vocii.

Un alt avantaj este că reţelele neuronale nu necesită programe puternice ci mai degrabă sînt rezultatul unor antrenamente asupra unui set de date, reţeaua învăţînd să scoată la ieşire rezultatele dorite. O dată ce a fost antrenată, reţeaua va fi capabilă să lucreze şi cu intrări diferite faţă de cele cu care a fost antrenată, atîta timp cît aceste intrări nu sînt foarte diferite. Acesta este un mare avantaj faţă de software-ul clasic care trebuie construit special pentru a opera cu fiecare intrare particulară în parte. Aceasta ne va permite să eliminăm probabil catastrofa virgulei lipsă.

Fractali şi sisteme de funcţii iterate

Formele fractale sînt clase de obiecte care rezultă din evoluarea repetată a unei funcţii simple, matematice, pentru a produce o formă complexă, cu un nivel de detaliu infinit. (O definiţie completă ne oferă Mandelbrot: Un fractal este o mulţime a cărei dimensiune Hausdorff nu este întreagă).

Descoperită de un matematician, tehnica fractalilor s-a dovedit nepreţuită în construcţia imaginilor pe calculator a obiectelor naturale, cu detalii şi texturi ciudate. În ultimii ani, au ieşit la iveală legăturile adînci dintre fractali, sistemele de funcţii iterate (IFS) şi dinamicile haotice, rezultatele fiind atît din domeniul biologiei cît şi din cel al informaticii.

În ştiinţele naturale, metodele fractale sînt utile pentru vizualizarea dinamicii haotice a sistemelor neliniare. Ştiinţa haosului, cunoscută şi ca dinamică neliniară, este o tehnică de modelare matematică, folosită pentru reprezentarea unor stări complexe ale sistemelor cu buclă de reacţie (feed back) de la inima umană pînă la sistemul vremii, interacţiile dintre neuronii mamiferelor şi mişcarea planetelor.

Una dintre cele mai utile aplicaţii ale fractalilor a fost, în industria PC-urilor, comprimarea imaginilor scanate după teoria IFS-urile oferă o metodă compactă de a reprezenta o subclasă de fractali, care poate fi partiţionată. Teoria lui Barnsley utilizează caracteristicile de partiţionare iterativă ale IFS-urilor pentru a obţine un factor de comprimare de 500:1 în unele cazuri, în altele de 10000:1.

Sisteme "L"

Chiar dacă sistemele fractale de ecuaţii pot produce imagini ale obiectelor naturale cum ar fi copacii sau munţii, cu o stranie asemănare, este greu de depistat legătura directă între funcţia $Z^2 + C \rightarrow Z$ şi un organism natural. Descoperitorul fractalilor, Benoit Mandelbrot, nu s-a preocupat de aceasta pentru că el n-a încercat să modeleze niciodată direct modul de lucru intern al proceselor biologice.

În contrast, sistemele lui Lindenmayer, sau sitemele "L", similare fractalilor sub anumite aspecte, au fost create cu scopul explicit de a modela natura. Aristid Lindenmayer, un biolog, a dezvoltat o teorie matematică a creşterii plantelor în 1968. Un sistem "L" este o mulţime de reguli care specifică o secvenţă repetată de transformări ale unei forme iniţiale. Regulile nu diferă faţă de cele BNF care specifică sintaxa limbajelor de programare, dar în loc de a analiza expresii, ele generează date. Dacă se dă o formă potrivită de pornire, şi regulile, se pot genera imagini ale unor obiecte ca nişte plante, care sînt incredibil de asemănătoare cu obiectul adevărat. Ca şi la fractali, un volum mic de informaţie, poate produce un obiect detaliat. Se poate spune că un sistem "L" este o altă modalitate de a reprezenta o mulţime fractală. Dar spre deosebire de fractalii convenţionali, sistemele "L" pot fi utilizate pentru a mima rezultatul final ca şi diferitele stadii de creştere ale obiectului generat.

Raţia mare de comprimare din sistemele "L" (şi din IFS-uri) ne dă o idee asupra modului în care o porţiune de 10^8 biţi ai ADN-ului poate specifica o reţea neuronală umană de 10^{11} elemente de procesare şi 10^{14} conexiuni. Nici un biolog nu a găsit locul în interiorul unei plante unde regulile sistemului "L" ar putea fi memorate. Nici nu s-a dorit aceasta. Aceste sisteme "L" pot fi văzute mai repede ca un exerciţiu formal pentru a înţelege cum poate fi modelat, specificat, prezis şi controlat procesul natural.

Recent, oamenii ce se ocupă cu grafica pe calculator, ca de exemplu Przemyslaw Prusinkiewicz au folosit sistemele "L" ca pe o tehnică de zidire a unor imagini naturale.

Algoritmi genetici

O tehnică inspirată biologic ce nu se foloseşte în grafica pe calculator sînt algoritmii genetici. Inventaţi de John Holland în 1975. Pe

scurt, această abordare foloseşte programele pentru a găsi o soluţie particulară într-un spaţiu general de căutare, prin mimarea procesului natural de evoluţie, mutaţie şi selecţie naturală. Deşi algoritmii genetici sînt inspiraţi direct din procesele biologice în practică legătura e aproape pierdută. Algoritmii genetici sînt un model rafinat de evoluţie aşa cum o reţea neuronală artificială este un model al creierului nu neapărat adevărat.

Dar o imitaţie perfectă a naturii nu este necesară. Aceste tehnici de rezolvare a problemelor sînt valoroase în sine.

Există o altă tehnică de căutare semialeatoare numită "călire simulată", (echivalentă la bază cu algoritmii genetici), care mimează cristalizarea unui lichid în răcire (sau călirea unui metal încălzit şi apoi răcit). Această tehnică este similară cu algoritmii genetici; prin generarea repetată a unei soluţii de încercare testînd-o apoi cu ţelul dorit, apoi se mută soluţia semialeatar pentru a vedea dacă o soluţie mai bună este disponibilă. Pe parcurs ce temperatura scade, programul converge spre o soluţie a problemei aproape optimă. Echivalenţa dintre călirea simulată şi algoritmii genetici vine din modul în care acelaşi proces se poate manifesta în acelaşi timp în sisteme biologice şi nebiologice.

Algoritmii genetici ai lui Holland lucrează acum cu o tehnică îmbunătăţită de căutare paralelă şi optimizare numită "sisteme clasificatoare". Sistemele clasificatoare sînt similare cu sistemele "L" prin faptul că sînt bazate pe reguli. De asemenea, ele încorporează evoluţia genetică şi mutaţia. S-a demonstrat că ele sînt formal echivalente cu sistemele conexioniste (reţele neuronale artificiale).

Alţi cercetători au combinat algoritmii genetici cu alte tehnici - de exemplu au utilizat algoritmii genetici pentru a dezvolta modele noi de reţele neuronale. Sau au utilizat algoritmii genetici pentru a transforma programele convenţionale - de exemplu programe de sortare aproape optimale pentru un anumit set de date. Ca şi reţelele neuronale, algoritmii genetici se potrivesc foarte uşor cu hardware-ul paralel, ceea ce explică interesul unor producători mari de semiconductori (Intel, TI) pentru reţelele neuronale şi alte tehnici bioinformatice.

Alte tehnici bioinformatice

Logica fuzzy, sau teoria mulţimilor fuzzy este o tehnică inspirată din natură, fără a încerca să fie un model realist al unui obiect fizic. Inventată de Lofti Zadeh în 1965, logica fuzzy este o extensie a logicii matematice care permite valori noi, între valorile tari de adevărat şi fals (1 şi 0). Intenţia este de a fi capabili să lucrăm într-un mod inteligibil cu noţiuni sau concepte imprecise, care nu au margini exacte. Aceasta nu mimează modul în care lucrează creierul uman, dar pare să se apropie de modul în care acţionează gîndirea umană.

Logica fuzzy şi-a găsit mulţi aderenţi în companiile japoneze. Această tehnologie este acum un element cheie în produse ca: aparatele de fotografiat Cannon cu autofocalizare, maşinile de spălat Hitachi, transmisiile Nissan şi Subaru, la fel ca şi în recunoaşterea scrisului de mînă la calculatorul Sony Palmtop.

Partea cealaltă a bioinformaticii

Pînă acum ne-am focalizat atenţia pe modurile în care se răsfrînge biologia în programe, dar o discuţie despre bioinformatică nu poate fi completă fără a menţiona cealaltă parte, utilizarea calculatoarelor şi a informaticii pentru a studia sistemele biologice, adică biologia computaţională. Cîteva din tehnicile menţionate anterior (reţeleie neuronale, sistemele "L") au pornit de la încercarea de înţelegere a sistemelor biologice. Dar tehnicile au devenit deja o tehnică de calcul depărtîndu-se de modelul biologic.

Cercetătorii rămîn să studieze în continuare bioorganismele ca unităţi de procesare a informaţiei. De exemplu, un autor în domeniul imunologiei face observaţia că sistemul imunologic, care este o reţea complexă de elemente interactive distribuite în corpul uman (aşa ca filamentul de trei picioare lungime al ADN-ului uman), pot fi văzute ca procese cognitive, entităţi capabile de căutare paralelă distribuită, recunoaşterea formelor şi memorie asociativă, nu altfel decît o reţea neuronală artificială sau un sistem clasificant genetic.

Biologia computaţională, a avut o influenţă largă în domeniul geneticii moleculare medicale, chiar dacă cercetătorii nu utilizează această frază în descrierea muncii lor. Ştiind că bioorganismele au rezultatul execuţiei de programare biochimice memorat pe o bandă codificată digital, (secvenţa ADN-ului genetic) cercetătorii medicali au descoperit că anumite boli sînt erori de transmisie date, în semnalul genetic. De exemplu, fibroza cistică dezordonată ereditară, vine dintr-o singură schimbare în valorile datelor, exact ca şi virgula lipsă din programele navetei spaţiale. Ca şi acea virgulă, eroarea poate fi detectată, corectată, programul fiind însănătoşit.

Natural, cele mai bune scule au fost construite pentru a îmbunătăţi aceste metode: secvenţiale de ADN controlate automat pe calculator (dezasamblare), maşini de reacţii înlănţuite polimerice, (duplicator de benzi digitale), staţii de lucru pentru proiectare moleculară, şi aşa mai departe. Există desigur aici şi probleme bioetice care trebuiesc discutate, la fel ca şi marile provocări tehnice. Şi, cine ştie, poate peste treizeci de ani, Borland sau Multiscope vor ieşi pe piaţă cu un debugger interactiv pentru *"forme-de-viaţă-dezvoltate-acasă"*, completat cu driver-e de citire/scriere pentru secvenţe ADN.

(E.R.)

(Prelucrare după articolul "What Is BioComputing?", Ray Valdés, Dr. Dobb's Journal #175, april 1991, p. 46)

CONSECO S.R.L.

cu sediul în Baia Mare , C.P.517, str. Siugariu 23/38, tel./fax: 40-94-36433

în colaborare cu

Societatea Comercială Româno - Canadiană

JM & CO. S.R.L.

din Bucureşti, str. Miletin nr. 1, sectorul 3, tel.:20.33.24, tel./fax : 20.27.05

vă oferă :

Calculatorul compatibil IBM-PC, format A4 Notebook, tip FALCON 320 N

cu următoarele caracteristici:

- **procesor 80386 SX, 20 MHz;**
- **soclu pentru 80387 SX;**
- **2 MB RAM pe placă, extensibil la 8 MB;**
- **1 x FDD 3,5" 1,44 MB (suportă şi 720 KB);**
- **HDD 40 MB (cu popsibilităţi de comandă pentru versiuni de 80 sau 120 MB);**
- **ecran LCD TSTM paper-white, 64 tonuri de gri, VGA 640x480;**
- **tastatură 82 taste cu 12 taste funcţionale şi tastatură numerică inclusă;**
- **porturi : 2 seriale, 1 paralel, conectori pentru FDD, monitor VGA color şi tastatură normală exterioară;**
- **sistem de operare DR DOS 5;**
- **baterii (acumulatori) NiCd incluse, aprox. 2 ore funcţionare continuă fără încărcare;**
- **alimentator pentru reţea c.a. 90 - 264 V;**
- **dimensiuni : 298 x 248 x 53mm.**
- **greutate (inclusiv bateriile): 3,2 Kg.**

Preţul este de **2100 $** *SUA .*

Toate preţurile includ atît vama cît şi ICM., care sînt achitate de către furnizor.

Plata se efectuează în lei la cursul zilei plăţii.

Memento: Funcţii Paradox 3.5

ABS	Valoarea absolută a unui număr ABS(Număr)
ACOS	Arccosinusul unui număr ACOS(Număr)
ARRAYSIZE	Dimensiunea unui tablou ARRAYSIZE(Tablou)
ASC	Returnează codul ASCII al unui caracter ASC(Caracter)
ASIN	Arcsinusul unui număr ASIN (Număr)
ATAN	Arctangenta unui număr ATAN (Număr)
ATAN2	Arctangenta în patru cadrani ATAN2 (Număr1, Număr2)
ATFIRST	Este articolul curent primul articol? ATFIRST ()
ATLAST	Este articolul curent ultimul articol? ATLAST ()
BANDINFO	Linia curentă în generatorul de rapoarte BANDINFO ()
BLANKDATE	Returnează o dată vidă BLANKDATE ()
BLANKNUM	Returnează un număr vid BLANKNUM ()
BOT	Testează încercarea de poziţionare înainte de începutul unei tabele BOT ()
CAVERAGE	Calculează media valorilor dintr-o coloană CAVERAGE (NumTabelă, NumCîmp)
CCOUNT	Numără valorile dintr-o coloană CCOUNT (NumTabelă, NumCîmp)
CHAR-WAITING	Aşteaptă un caracter în buffer-ul tastaturii? CHARWAITING ()
CHECKMARK-STATUS	Cîmpul curent este marcat în cerere? CHECKMARKSTATUS ()
CHR	Converteşte un cod ASCII în caracterul corespunzător CHR (Număr)
CMAX	Valoarea maximă dintr-o coloană CMAX (NumTabelă, NumCîmp)
CMIN	Valoarea minimă dintr-o coloană CMIN (NumTabelă, NumCîmp)
CNPV	Valoarea netă a valorilor unei coloane CNPV (NumTabelă, NumCîmp, Număr)
COL	Coloana în care este poziţionat cursorul pe ecranul utilizator PAL COL ()
COLNO	Numărul coloanei curente din imaginea aflată în spaţiul de lucru COLNO ()
COS	Cosinusul unui unghi COS (Număr)
CSTD	Deviaţia standard a valorilor dintr-o coloană CSTD (NumTabelă, NumCîmp)
CSUM	Suma valorilor dintr-o coloană CSUM (NumTabelă, NumCîmp)
CURSOR-CHAR	Caracterul aflat în poziţia în care se găseşte cursorul CURSORCHAR ()
CURSORLINE	Textul de pe linia curentă a spaţiului de lucru CURSORLINE ()
CVAR	Variaţia valorilor dintr-o coloană CVAR (NumTabelă, NumCîmp)
DATEVAL	Converteşte un şir într-o dată DATEVAL (Şir)
DAY	Data unei zile ca şi număr DAY (Dată)
DIRECTORY	Numele directorului curent DIRECTORY ()
DIREXISTS	Există directorul specificat? DIREXISTS (CaleDOS)
DOW	Returnează ziua săptămînii ca şir DOW (Dată)
DRIVESPACE	Spaţiul rămas pe disc DRIVESPACE (Şir)
DRIVE-STATUS	Unitatea este pregătită? DRIVESTATUS (Şir)
EOT	Testează încercarea de poziţionare după sfîrşitul unei tabele EOT ()
ERRORCODE	Codul corespunzător erorii de execuţie celei mai recente ERRORCODE ()
ERROR-MESSAGE	Mesajul corespunzător erorii de execuţie celei mai recente ERRORMESSAGE ()
ERRORUSER	Numele urilizatorului care a blocat un obiect ERRORUSER ()
EXP	Exponenţiala unui număr EXP (Număr)
FAMILY-RIGHTS	Verifică drepturile de acces la familia unei tabele FAMILYRIGHTS (NumTabelă, Şir)
FIELD	Numele cîmpului curent FIELD ()
FIELDINFO	Returnează indicatorul cîmpului curent în modul Report sau Form FIELDINFO ()
FIELDNO	Poziţia unui cîmp într-o tabelă FIELDNO (NumCîmp, NumTabelă)
FIELDRIGHTS	Verifică drepturile de acces la un cîmp dintr-o tabelă FIELDRIGHTS (NumTabelă, NumCîmp, Şir)
FIELDSTR	Valoarea curentă a unui cîmp ca şir FIELDSTR ()
FIELDTYPE	Tipul datelor din cîmpul curent FIELDTYPE ()
FILESIZE	Dimensiunea unui fişier în bytes FILESIZE (NumFişier)
FILL	Returnează un şir de lungime specificată, format din aparîţiile repetate ale unui caracter FILL (Expresie, Număr)

FORM	Returnează numele formei curente pentru tabela curentă FORM()
FORMAT	Formatează o expresie FORMAT (SpecFormat, Şir)
FORMTYPE	Verifică tipul formei curente FORMTYPE (Şir)
FV	Valoarea urnătoare dintr-o serie de "cash flows" FV (Număr1, Număr2, Număr3)
GETCHAR	Citeşte un caracter de la tastatură GETCHAR ()
GRAPHTYPE	Tipul graficului curent GRAPHTYPE ()
HELPMODE	Tipul de help curent HELPMODE ()
IMAGEC-AVERAGE	Media valorilor din coloana curentă a imaginii curente IMAGEAVERAGE ()
IMAGEC-COUNT	Afişează numărul de valori din coloana curentă a imaginii curente IMAGECCOUNT ()
IMAGECMAX	Valoarea maximă din coloana curentă a imaginii curente IMAGECMAX ()
IMAGECMIN	Valoarea minimă din coloana curentă a imaginii curente IMAGECMIN ()
IMAGECSUM	Suma valorilor din coloana curentă a imaginii curente IMAGECSUM ()
IMAGENO	Poziţia imaginii curente din spaţiul de lucru IMAGENO ()
IMAGETYPE	Tipul imaginii curente IMAGETYPE ()
INT	Partea întreagă a unui număr INT (Număr)
ISASSIGNED	A fost asignată o variabilă sau un element de matrice? ISASSIGNED (Şir)
ISBLANK	Expresia este vidă? ISBLANK (Expresie)
ISBLANK-ZERO	Valorile vide vor fi tratate în calcule ca şi zero? ISBLANKZERO ()
ISEMPTY	Tabela este goală? ISEMPTY (NumTabelă)
ISENCRYPTED	Tabela este protejată prin parolă? ISENCRYPTED (NumTabelă)
ISFIELDVIEW	Ne aflăm în modul Field View? ISFIELDVIEW ()
ISFILE	Există fişierul specificat? ISFILE (Şir)
ISFORMVIEW	Tabela curentă este afişată printr-o formă? ISFORMVIEW ()
ISINSERT-MODE	Paradox este în modul de inserare? ISINSERTMODE ()
ISLINK-LOCKED	Tabela curentă est blocată prin legare? ISLINKLOCKED ()
ISMULTI-FORM	Forma tabelei curente conţine forme şi pentru alte tabele? ISMULTIFORM (NumTabelă, NumFormă)
ISMULTI-REPORT	Raportul conţine tabele legate? ISMULTIREPORT (NumTabelă,NumRaport)
ISRUNTIME	Programul este executat de versiunea Paradox Runtime? ISRUNTIME ()
ISSHARED	Tabela se află într-un director partajat? ISSHARED (NumTabelă)
ISTABLE	Există tabela? ISTABLE (NumTabelă)
ISVALID	Are fişierul un conţinut valid? ISVALID ()
LEN	Lungimea unui şir LEN (Expresie)
LINKTYPE	Tipul de legătură dintre tabele aparţinînd unei forme multitabelare LINKTYPE ()
LN	Logaritmul natural al unui număr LN (Număr)
LOCKSTATUS	Cîte blocări de un anumit tip au fost plasate asupra tabelei curente? LOCKSTATUS (NumTabelă, TipBlocare)
LOG	Logaritmul în baza 10 al unui număr LOG (Număr)
LOWER	Converteşte un şir în litere mici LOWER (Expresie)
MATCH	Compară un şir cu un şablon MATCH (Şir, Şir pattern [,NumListăVariabile])
MAX	Maximul dintre două numere MAX (Număr1, Număr2)
MEMLEFT	Memorie RAM disponibilă unei aplicaţii MAMLEFT ()
MENUCHOICE	Alegerea curentă din meniul curent Paradox MENUCHOICE ()
MIN	Minimul dintre două numere MIN (Număr1, Număr2)
MOD	Restul împărţirii unui număr cu altul MOD (Număr1, Număr2)
MONITOR	Tipul monitorului folosit MONITOR ()
MONTH	Luna unei date ca un număr MONTH (Dată)
MOY	Luna unei date ca şir MOY (Dată)
NETTYPE	Ce fel de reţea este folosită? NETTYPE ()
NFIELDS	Numărul cîmpurilor dintr-o tabelă NFIELDS (NumTabelă)
NIMAGERE-CORDS	Numărul de articole legate sau nu din imaginea curentă NIMAGERECORDS ()
NIMAGES	Numărul de imagini din spaţiul de lucru NIMAGES ()
NKEYFIELDS	Numărul de cîmpuri cheie dintr-o tabelă NKEYFIELDS (NumTabelă)
NPAGES	Numărul de pagini dintr-un raport sau dintr-o formă NPAGES ()
NRECORDS	Numărul de articole dintr-o tabelă NRECORDS ()

NROWS	Numărul liniilor din imaginea sau raportul curent NROWS ()
NUMVAL	Converteşte un şir într-un număr NUMVAL (Şir)
PAGENO	Numărul paginii curente dintr-un raport sau dintr-o formă PAGENO ()
PAGEWIDTH	Lăţimea paginii din raportul curent PAGEWIDTH ()
PI	Returnează valoarea pi PI ()
PMT	Calculează rata de amortizare PMT (Număr1, Număr2, Număr3)
POW	Ridică un număr la o putere POW (Număr1, Număr2)
PRINTER-STATUS	Imprimanta este pregătită? PRINTERSTATUS ()
PRIVDIR	Numele directorului privat al reţelei utilizatorului PRIVDIR ()
PV	Valoarea actuală a unei serii de plăţi PV (Număr1, Număr2, Număr3)
QUERY-ORDER	Ordinea implicită a cîmpurilor din tabela de răspuns QUERYORDER ()
RAND	Returnează un număr aleator RAND ()
RECNO	Numărul articolului curent RECNO ()
RECORD-STATUS	Verifică starea articolului curent RECORDSTATUS (Şir)
RETRY-PERIO	Care este valoarea perioadei de reîncercare? RETRYPERIOD ()
RMEMLEFT	Returnează nivelul de memorie disponibil RMEMLEFT ()
ROUND	Rotunjeşte un număr ROUND (Număr1, Număr2)
ROW	Linia în care este poziţionat cursorul pe ecranul utilizator PAL ROW ()
ROWNO	Afişează numărul curent al rîndului dintr-o imagine, raport sau formă ROWNO ()
SDIR	Numele căii programului curent SDIR ()
SEARCH	Caută un subşir într-un şir SEARCH (Subşir, Şir)
SIN	Sinusul unui unghi SIN (Număr)
SORTORDER	Ce fel de ordine de sortare a fost instalată? SORTORDER ()
SPACES	Returnează un şir de spaţii SPACES (Număr)
SQRT	Rădăcina pătrată a unui număr SQRT (Număr)
STRVAL	Converteşte o expresie într-un şir STRVAL (Expresie)
SUBSTR	Returnează un subşir dintr-un şir SUBSTR (Şir, Număr1, Număr2)
SYSCOLOR	Care sînt culorile curente ale sistemului Paradox? SYSCOLOR (Număr)
SYSMODE	Modul curent Paradox SYSMODE ()
TABLE	Numele tabelei curente TABLE ()
TABLE-RIGHTS	Verifică drepturile de acces la o tabelă Paradox TABLERIGHTS (NumTabelă, Şir)
TAN	Tangenta unui unghi TAN (Număr)
TIME	Ora zilei TIME ()
TODAY	Data zilei de astăzi TODAY ()
TYPE	Tipul unei expresii TYPE (expresie)
UPPER	Converteşte un şir în caractere mari UPPER (Expresie)
USERNAME	Care este numele utilizatorului curent? USERNAME ()
VERSION	Afişează numărul de versiune Paradox folosit VERSION ()
WINDOW	Returnează un şir ce include conţinutul ferestrei de mesaje a Paradox-ului WINDOW ()
YEAR	Anul unei date YEAR (Dată)

Taste funcţionale

<F1>	Help
<F2>	Do-It
<F3>	Up Image
<F4>	Down Image
<F5>	Example
<F6>	Checkmark
<F7>	Form Toggle
<F8>	Clear Image
<F9>	Edit
<F10>	Menu (Meniu)

<Alt>/<F3>	Instant Script Record
<Alt>/<F4>	Instant Script Play
<Alt>/<F5>	Field View
<Alt>/<F6>	Check Plus
<Alt>/<F7>	Instant Report
<Alt>/<F8>	Clear All
<Alt>/<F9>	Coedit
<Ctrl>/<F6>	Check Descending
<Ctrl>/<F7>	Instant Graph
<Shift>/<F6>	Group By

Combinaţii speciale de taste

<Alt>/<C>	**Color Palette** - Comută paleta de culori on/off în timpul proiectării unei forme
<Ctrl>/<D>	**Ditto** - La editare, sau introducere date, copiază valoarea aceluiaşi cîmp din articolul anterior (duplicare valoare)
<Ctrl>/<F>	**Field View** - Mută cursorul în interiorul unui cîmp permiţînd editarea lui (la fel ca <Alt>/<F5>)
<Alt>/<K>	**Key Viol** - Vizualizează articolele la care apar situaţii conflictuale datorate unor duplicări de chei
<Ctrl>/<L>	**Resync** - Resincronizează articolele într-o formă multitabelară
<Alt>/<L>	**Lock Toggle** - Blochează sau deblochează un articol
<Ctrl>/<O> <Alt>/<O>	**DOS** - Suspendă temporar sesiunea Paradox şi predă controlul la nivelul sistemului de operare DOS. Pentru revenire în Paradox trebuie tastat **Exit**
<Ctrl>/<R>	**Rotate** - Roteşte coloanele unei tabele
<Alt>/<R>	**Refresh** - Actualizează imaginea afişată pe ecran cu modificările efectuate în tabelă de utilizatorii din reţea
<Ctrl>/<U>	**Undo** - Reface incremental modificările făcute într-o tabelă într-unul din modurile Edit, Coedit sau DataEntry
<Ctrl>/<V>	**Vertical Ruler Toggle** - Afişează sau maschează contorul liniilor în timpul editării unui raport sau a scrierii unui program
<Alt>/<X>	**CrossTab** - Face referinţele încrucişate pentru imaginea curentă
<Ctrl>/<Y>	**Delete Line** - Şterge linia curentă la editarea unui raport sau la scrierea unui program
<Ctrl>/<Z>	**Zoom** - Localizează prima apariţie a unei valori într-o tabelă
<Alt>/<Z>	**Zoom Next** - Localizează următoarea apariţie a unei valori într-o tabelă
<Ctrl>/<Backspace>	**"Turbo" Backspace** - Şterge conţinutul cîmpului curent în mod Edit sau DataEntry
<Tab> <Shift>/<Tab>	**Tab** - Deplasare între cîmpuri în mod vizualizare tabelă şi între coloane în generatorul de rapoarte.

Obiecte Paradox	
Extensie	**Tipul obiectului**
.DB	Tabelă
.F sau .F??	Formă
.R sau .R??	Format raport
.SET	Setări imagine
.VAL	Marcaje de validare
.PX	Index primar
.X??	Index secundar
.Y??	Index secundar
.SC	Script (program)
.G	Specificaţii grafic
Semnul ? înlocuieşte un singur caracter în extensie.	

Specificaţii de format pentru ieşire date			
Tip format	Semnificaţie		Tip date
Dimensiune	Wn	Stabileşte dim. permisă	toate
	Wn.m	Setează dim. şi nr. poz. zec.	N,$
Aliniere	AL	Aliniere la stînga	toate
	AR	Aliniere la dreapta	toate
	AC	Centrează valoarea	toate
Caractere	CU	Converteşte în litere mari	toate
	CL	Converteşte în litere mici	toate
	CC	Converteşte în forma iniţială	toate
Editare	E$	Include semnul $	N,S,$
	EC	Include virgula ca separator	N,S,$
	EZ	Afişează zerourile nesemnif.	N,S,$
	EB	Foloseşte spaţiu în locul zerourilor nesemnificative	N,S,$
	ES	Afişează folosind notaţia ştiinţifică	N,S,$
	E*	Foloseşte * pentru zerourile nesemnificative	N,S,$
	EI	Foloseşte formatul internaţional pentru numere	N,S,$
Semn	S+	Afişează semnul + sau -	N,S,$
	S-	Afişează semnul -	N,S,$
	SP	Afişează val. negative în ()	N,S,$
	SD	Afişează notaţia DB sau CR	N,S,$
	SC	Afişează CR după valorile negative	N,S,$
Date	D1	Format: ll/zz/aa	D
	D2	Format: Luna zz, aaaa	D
	D3	Format: ll/zz	D
	D4	Format: ll/aa	D
	D5	Format: zz-lll-aa	D
	D6	Format: lll zz	D
	D7	Format: zz-lll-aaaa	D
	D8	Format: ll/zz/aaaa	D
	D9	Format: zz.ll.aa	D
	D10	Format: zz/ll/aa	D
	D11	Format: aa-ll-zz	D
Logic	LY	Utilizează Yes/No pentru True/False	L
	LO	Utilizează On/Off pentru True/False	L
Tip date: N = Numeric, S = numeric scurt, $ = Valoare, D = Dată, L = Valoare logică			

Paradox este un S.G.B.D. al firmei:
Borland International Inc.
P.O. Box 660001, Scotts Valley, CA 95067-0001
Reprezentant autorizat în România:
S.C. Logic - Sibiu, Calea Dumbrăvii 21
tel. (924) 46652, 45475, fax: (924) 46704
- Bucureşti, b-dul Tineretului 8-10
tel. (90) 754900, fax: (90) 757135

Cine indexează mai bine ?

Nu, nu e vorba de salarii (din păcate...) ci de fişiere. Ştiut fiind că în general orice aplicaţie serioasă nu poate să se descurce fără fişiere secvenţial-indexate (sau ceva similar), vă prezentăm un program - scris şi testat sub Microfocus Cobol v1.2 - pe care vă rugăm să-l rescrieţi în alt limbaj, într-o versiune care să se apropie cît mai mult de cea pe care v-o prezentăm. Scopul - obţinerea unor cifre relative la viteza de execuţie, care să permită o opţiune pentru cine porneşte să construiască o aplicaţie.

Am decis să 'lansăm' ca model un program Cobol pe considerentul că este uşor de citit de către orişicine. Desigur, acest test nu poate oferi date decisive pentru orice situaţie. Dar informaţii pertinente sperăm să putem sintetiza, cu ajutorul dumneavoastră. Vă rugăm să ne remiteţi nu numai programul sursă şi executabil, ci şi date relative la calculatorul pe care aţi rulat şi timpii obţinuţi. Varianta prezentată, scrisă şi rulată de Man Vasile, de la Oficiul de Calcul al D.Tc. Mureş, pe un 286/15 MHz, cu un harddisk de 40 Mo cu un timp mediu de acces de 28 ms, cu 1 Mo RAM, lansată de sub XTGold cu buffers=35 şi files=50 în CONFIG.SYS, fără vreun program cache activ, a avut nevoie de 1h30'58,44" pentru generare, 2'50,6" pentru prima consultare şi 1'33,2" pentru a doua consultare. Fişierul de test ocupă 3,4 Mo, indexul 2,5 Mo iar programul executabil cca 111,5 ko. În aceleaşi condiţii, dar sub Cobol 2.0, (care altfel se pare că uneori are ciudăţenii) timpii sînt 8'54,65", 1'17,04" şi 36,03", iar dimensiunile 3,6 Mo şi respectiv 2,5 Mo.

Iosif Fettich

```
1     IDENTIFICATION DIVISION.
2     PROGRAM-ID. DEMCOBIF.
3     * REMARKS. PROGR. DEMCOBIF CREAZA UN FIS. SECV.-INDEXAT
4     * IN ACCES DIRECT DE CCA 100000 ARTICOLE APOI INTEROGHEA-
5     * ZA ÎN ACCES DIRECT 1000 ARTICOLE IN DOUA MODALITATI.
6     ENVIRONMENT DIVISION.
7     CONFIGURATION SECTION.
8     INPUT-OUTPUT SECTION.
9     FILE-CONTROL.
10     SELECT DEMCOBIF ASSIGN TO "DEMCOB" ORGANIZATION INDEXED
11            ACCESS RANDOM RECORD KEY ABACT.
12    DATA DIVISION.
13    FILE SECTION.
14    FD  DEMCOBIF    LABEL RECORD STANDARD
15                BLOCK CONTAINS 8 RECORDS.
16    01  ARTACT.
17     02 ABACT.
18      03  AB1 PIC 9(6).
19      03  AB2 PIC X(2).
20     02 ALTC1 PIC X(6).
21     02 ALTC2 PIC Z(5)9.
22     02 ALTC3 PIC X(6).
23     02 ALTC4 PIC Z(5)9.
24    WORKING-STORAGE SECTION.
25     77  I  PIC 9(6) COMP.
26     77  J  PIC 9(6) COMP.
27     77  K  PIC 9(6) COMP VALUE 0.
28     77  L  PIC 9(4) COMP VALUE 0.
29     77  REST PIC 9(3) COMP.
30     77  SECVSF PIC 9(6) COMP VALUE 100000.
31     77  SCHIMB PIC XX.
32     01  WDATA-EXEC.
33      02  MES1 PIC X(22) VALUE '  INDEXARE INCEPUTA  '.
34      02  FILLER PIC X(17) VALUE ' - (HHMMSS..) : '.
35      02  HMS PIC 99B99B99B99.
36     01  WCH-GEN.
37      02  WCH-G1 PIC X(54) VALUE
                  'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghij
38    -             'klmnopqrstuvwxyz  '.
39      02  WCH-G2 REDEFINES WCH-G1 PIC XX OCCURS 27.
40     01  WCHEI-CITIRE.
41      02  WCH-CIT1  PIC X(8) OCCURS 1000.
42    PROCEDURE DIVISION.
43    INCEP.
44        OPEN OUTPUT DEMCOBIF
45        DISPLAY '          P R O G R A M    D E M C O B I F'
46        MOVE SPACES TO WCHEI-CITIRE
47        MOVE 'INF.GR1' TO ALTC1
48        MOVE 'INF.GR2' TO ALTC3
49        ACCEPT HMS FROM TIME
50        DISPLAY WDATA-EXEC.
51        PERFORM GEN-FIS VARYING I FROM 1 BY 1 UNTIL I > 50000
52        ACCEPT HMS FROM TIME
53        MOVE '  INDEXARE TERMINATA  ' TO MES1
54        DISPLAY WDATA-EXEC.
55        CLOSE DEMCOBIF
56        OPEN INPUT DEMCOBIF
57        ACCEPT HMS FROM TIME
58        MOVE 'INTEROGARE 1 INCEPUTA' TO MES1
59        DISPLAY WDATA-EXEC.
60        MOVE 1001 TO J
61        PERFORM CIT1-FIS VARYING I FROM 1 BY 1 UNTIL I > 500
62        ACCEPT HMS FROM TIME
63        MOVE 'INTEROGARE 1 TERMINATA' TO MES1
64        DISPLAY WDATA-EXEC.
65        ACCEPT HMS FROM TIME
66        MOVE 'INTEROGARE 2 INCEPUTA' TO MES1
67        DISPLAY WDATA-EXEC.
68        PERFORM CIT2-FIS VARYING I FROM 1 BY 1 UNTIL I > 1000
69        ACCEPT HMS FROM TIME
70        MOVE 'INTEROGARE 2 TERMINATA' TO MES1
71        DISPLAY WDATA-EXEC.
72        CLOSE DEMCOBIF
73        STOP RUN.
74    GEN-FIS.
75        SUBTRACT 1 FROM SECVSF
76        DIVIDE 26 INTO I GIVING K REMAINDER REST
77        PERFORM GEN-CH
78        MOVE I TO AB1 ALTC2 ALTC4
79        PERFORM SCR-FIS
80        DIVIDE 26 INTO SECVSF GIVING K REMAINDER REST
81        PERFORM GEN-CH
82        MOVE SECVSF TO AB1 ALTC2 ALTC4
83        PERFORM SCR-FIS.
84    GEN-CH.
85        ADD 1 TO REST
86        MOVE WCH-G2 (REST) TO AB2
87        MOVE WCH-G2 (REST) TO SCHIMB
88        MOVE WCH-G2 (27) TO WCH-G2 (REST)
89        MOVE SCHIMB TO WCH-G2 (27).
90    SCR-FIS.
91        ADD 1 TO J
92        IF J > 95
93           MOVE 0 TO J
94           ADD 1 TO L
95           IF L < 1001
96              MOVE ABACT TO WCH-CIT1 (L).
97        WRITE ARTACT   INVALID KEY
98                DISPLAY 'CHEIE DUBLA / NECRESCATOARE' ABACT.
99    CIT1-FIS.
100       MOVE WCH-CIT1 (I) TO ABACT
101       READ DEMCOBIF    INVALID KEY
102                DISPLAY ' CHEIE NEGASITA ' ABACT.
103       SUBTRACT 1 FROM J
104       MOVE WCH-CIT1 (J) TO ABACT
105       READ DEMCOBIF    INVALID KEY
106                DISPLAY ' CHEIE NEGASITA ' ABACT.
107   CIT2-FIS.
108       MOVE WCH-CIT1 (I) TO ABACT
109       READ DEMCOBIF    INVALID KEY
110                DISPLAY ' CHEIE NEGASITA ' ABACT.
```

DEMCOBIF.CBL

Programele de educaţie Novell

Programul NAEC

NAEC reprezintă abrevierea de la Novell Authorized Education Center. Aceste centre primesc de la Novell informaţiile tehnice necesare în procesul de pregătire a viitorilor experţi în utilizarea produselor firmei.

Pentru ca un centru de pregătire să devină un NAEC el trebuie să îndeplinească o serie de cerinţe extrem de exigente atît din punctul de vedere al dotării tehnice cît şi din punctul de vedere al pregătirii instructorilor din centrul respectiv.

Astfel instructorii trebuie în prealabil să urmeze cursuri autorizate Novell fiind supuşi apoi unei testări dure; doar după trecerea acestor teste candidatul devine CNI (Certified NetWare Instructors). În continuare proaspătul CNI trebuie să-şi reconfirme statutul susţinînd periodic noi examene.

Prin intermediul programului NAEC Novell îşi susţine programele de autorizare şi certificare; calitatea pregătirii asigurate de NAEC este controlată în amănunt de firma Novell, fiind considerată la fel de importantă ca şi calitatea celorlalte produse Novell.

Prin intermediul acestui program, Novell reuneşte distribuitori, reseller-i, OEM (Original Equipment Manufacturer), consultanţi centre de pregătire într-o alianţă care are drept scop final asigurarea calităţii procesului de pregătire în utilizarea optimă a produselor Novell.

În cadrul NAEC se desfăşoară două programe principale: CNE (Certified NetWare Engineer) şi CNI (Certified NetWare Instructor).

Programul CNE se referă la certificarea Novell a unor ingineri din cadrul diferitelor organizaţii. O organizaţie care are în cadrul ei un CNE se bucură de un sprijin deosebit în domeniul tehnic şi comercial din partea firmei Novell.

Certificarea CNE se obţine prin examene care urmează cursurilor absolvite într-un NAEC. Certificarea CNE este şi ea urmată de noi cursuri, de noi examene, pregătirea referindu-se de fiecare dată la cele mai noi evoluţii în domeniul reţelelor de calculatoare, evoluţii aflate de o bună bucată de timp în strînsă legătură cu Novell.

Pregătirea în cadrul NAEC este asigurată în mod obligatoriu de CNI. Un CNI se specializează în unul sau mai multe din domeniile următoare:

- specialist NetWare
- specialist în comunicaţii
- specialist în gestiunea reţelei
- specialist în dezvoltare
- specialist în legături Unix - TCP/IP

Cu începere din luna decembrie firma SCOP va pune la dispoziţia solicitanţilor primul NAEC din România, asigurînd participanţilor la cursuri o pregătire conformă celor mai exigente standarde internaţionale ale momentului.

Politica Novell în comerţul internaţional

Novell are în vedere următoarele canale de distribuţie pentru produsele sale:

- International Distributor. Reprezintă un reseller (revînzător) care comercializează şi distribuie produse numai prin intermediul unui Novell Authorized Reseller.

- International Remarketer. Reprezintă un Internaţional Distribuitor, International System Integrator sau International Value Added Reseller.

- International System Integrator. Reprezintă un reseller care oferă soluţii integrate şi care primeşte produsele Novell direct de la Novell.

- International Value Added Reseller. Reprezintă un reseller care oferă o soluţie la cheie, adresată unei anumite nişe a pieţei, soluţia constînd din produse software proprii la nivelul aplicaţiilor. Soluţia este implementată pe produse Novell.

- Novell Authorized Reseller. Reprezintă un reseller care cumpără produse de la International Remarketers şi le revinde către End Users (utilizatorii finali).

Novell acordă unei firme, în baza atentei analize a posibilităţilor tehnice dar şi comerciale, unul din gradele enumerate mai sus.

În cazul României, firma SCOP a fost investită ca International Remarketer. Această înţelegere permite firmei SCOP să comercializeze produse Novell din cadrul unei arii bine definite şi stabilite de obicei la nivelul frontierelor naţionale. În cadrul acestei arii, comercializarea produselor Novell se face prin intermediul unor Novell Authorized Resellers sau direct, dar numai strict în aria stabilită. Este cu desăvîrşire interzisă comercializarea de produse Novell utilizînd agenţi distribuitori, etc. În afara ariei stabilite. Dacă totuşi are loc o astfel de comercializare, Novell nu mai are nici un fel de obligaţie referitoare la garanţia produsului exportat, de asemenea se pierde licenţa pentru produsul respectiv, lucru care duce la pierderea oricărui suport tehnic din partea firmei Novell, ca şi, aspect mult mai grav, la imposibilitatea ca ulterior să se obţină versiunea următoare a produsului în schimbul unei sume reduse de bani (aşa-numitul upgrade).

În momentul de faţă, în România există mai multe firme care practică comercializarea unor produse Novell fără a avea dreptul să facă acest lucru; deci atunci cînd cumpăraţi un sistem de operare pentru reţeaua dvs. este foarte indicat să vă interesaţi de unde a obţinut furnizorul dvs. produsul respectiv.

În cazul în care nu l-a obţinut pe canalul Novell normal este preferabil să cîntăriţi mai atent decizia dvs. deoarece veţi pierde garanţia produsului, suportul tehnic Novell ca şi posibilitatea de upgrade. Mare atenţie la produsele Novell pirat deoarece achiziţia unui astfel de produs vă poate cauza în viitor mari neplăceri concretizate în amenzi foarte mari.

Şi un ultim sfat: dacă furnizorul dvs. declară că este distribuitor autorizat Novell, cereţi-i documentele doveditoare.

Oricum, firma SCOP vă stă cu plăcere la dispoziţie pentru orice informaţie referitoare la produsele Novell; nu vă rămîne decît să ne contactaţi telefonic sau să ne oferiţi plăcerea unei vizite.

ing. Sturzoiu Marius

NOVELL PRIN scop

NOVELL INTERNATIONAL AUTHORISED REMARKETER

Vă propune integrarea în sistemul autorizat de distribuţie în România.
Veţi beneficia astfel în mod oficial de:

- programul de upgrade al firmei NOVELL
- autorizarea firmei dvs. ca NOVELL AUTHORISED RESELLER
- suportul tehnic şi de marketing necesar.

IBM

EVEREX
EVER for EXcellence

COMPEX

Summagraphics / HOUSTON INSTRUMENT

scop

AUTHORISED REMARKETER IN ROMANIA

Telefon: 11.74.21 / 11.92.48. **Fax: 11.73.74.**

Noi sîntem soluţia !

Reţelele - o analiză a costurilor "la bani mărunţi"

Administratorii văd în primul rînd avantajele reţelelor şi consecinţa lor imediată, optimizarea proceselor de comunicaţie. Dar conducerea întreprinderilor şi contabilii, de cele mai multe ori, nu au acelaşi punct de vedere. Ei decid în funcţie de rentabilitate.

Fiecare întreprindere are colaboratori (angajaţi) care au idei nemaipomenite şi un simţ deosebit pentru tehnologii noi. Ei recunosc cu precizie strategiile actuale sau viitoare care ar putea optimiza procesul zilnic de lucru. Dar transpunerea acestor idei în practică eşuează de multe ori, deoarece aceşti angajaţi nu sînt în stare să evalueze costurile acestor noi tehnologii. Conducătorii întreprinderilor şi contabilii, care decid dacă un proiect se realizează sau nu, se gîndesc doar la finanţe. Urmarea:

Deseori strategiile de perspectivă nu se materializează, ceea ce duce la pierderea competitivităţii întreprinderii. O "viziune" tehnică nu este de ajuns. Trebuie demonstrat avantajul în afaceri pe care îl aduce fiecare bănuţ cheltuit.

În cazul introducerii unor înnoiri tehnologice, trebuie analizată influenţa acestui pas asupra întreprinderii. Trebuie făcută o analiză a costurilor.

Mulţi administratori evită analiza costurilor, deoarece sînt de părere că aceasta ar fi prea complicată. Totuşi problema este mult mai simplă decît pare la prima vedere. Cel mai important lucru este interpretarea corectă a cifrelor. Mai este nevoie, în afară de un soft de calcul tabelar, de puţină înţelegere pentru terminologia folosită în lumea finanţelor (vezi caseta).

Mijloace ajutătoare pentru analiza costurilor

După ce v-aţi familiarizat cu termenii şi concepţiile cele mai importante din "lumea investiţiilor", puteţi începe analiza. Există mai multe soluţii pentru calcularea costurilor unui proiect, dintre care două sînt cele folosite mai des: analiza valorii actuale şi analiza timpului de amortizare.

Valoarea curentă şi valoarea netă actuală

Cu siguranţă, una din cele mai răspîndite metode este cea a analizei valorii curente, numită şi "valoare în numerar". Aici se combină concepţiile taxei de scont, valorii banilor în timp şi creşterea în timp a valorii banilor.

Intervalul de timp pentru care se efectuează analiza va fi de pildă 2 ani. S-a aflat de la contabilitate că taxa actuală de scont este de 10% . La începutul anului 1991 s-au investit 10 mărci şi se aşteaptă ca prin introducerea proiectului să se economisească pînă la sfîrşitul anului 1991 5 mărci şi alte 5 mărci pînă la sfîrşitul anului 1992.

Deci analiza va arăta, mai întîi, astfel:

- Investiţie 1991: 10 mărci
- Economisire 1991: 5 mărci
- Economisire 1992: 5 mărci.

Formulele folosite pentru calcularea valorii în numerar oferă o metodă pentru calcularea pe de o parte a puterii de cumpărare viitoare a banilor şi pe de altă parte, a influenţei acesteia asupra fluxului de capital.

Cele mai multe programe de calcul tabelar conţin deja formula corespunzătoare; aceasta este:

Valoarea netă actuală = FCn/(1+d)a

- FCn = fluxul de capital în anul n (0 = anul în curs)
- d = taxa de scont
- a = numărul de ani pentru care se face calculul

Valoarea în numerar a investiţiei a fost de 10 mărci. Valoarea corespunzătoare a capitalului ce va intra sub formă de economii va fi de doar 8,68 mărci, cu toate că se economisesc în total 10 mărci pînă la sfîrşitul anului 1992 La această sumă se ajunge dacă economiile pronosticate pînă la sfîrşitul anului 1992 se convertesc în mărci "actuale", deci în valoare numerar.

Valoarea netă actuală se referă la diferenţa dintre întregul capital care intră şi cel care iese. Această valoare trebuie să fie în orice caz pozitivă, deci mai mare decît zero. Dacă valoarea netă actuală este negativă, proiectul nu este de dorit, deoarece el costă întreprinderea mai mult decît îi aduce.

Valoarea netă actuală din exemplul de mai sus este de 8,68 mărci minus 10 mărci investiţia, deci un minus de 1,32 mărci. La aceasta se mai adaugă costul banilor, adică taxa de scont. Ar trebui deci adăugat încă o marcă, reprezentînd 10% din investiţie. Prin urmare, proiectul ar fi respins, deoarece ar duce întreprinderea la un minus şi nici măcar valoarea investiţiei nu ar putea fi acoperită.

Timpul de reîntoarcere a capitalului (timpul de amortizare)

Un alt element ajutător pentru efectuarea unei analize a costurilor este calculul timpului în care capitalul investit se reîntoarce.

Timpul de reîntoarcere, denumit şi timp de amortizare, este numărul anilor necesari pentru recuperarea investiţiilor dintr-un proiect. Această metodă este uşor de înţeles şi implică calcule simple. Dacă de exemplu. la începutul anului 1991 s-au cheltuit 300 de mărci şi anual se recuperează 100 de mărci, atunci perioada de amortizare este de 3 ani.

Dacă această metodă este atît de simplă, se pune problema de ce să se mai folosească cea a valorii actuale. Metoda tim-

pului de amortizare are unele limitări. Ea nu ţine cont de valoarea în timp a banilor şi nici de fluxul de capital după prima investire. Cel mai bine se poate explica acest lucru cu un exemplu:

	Proiect A	Proiect B
Investiţii:	300DM	300DM
Economie 1991:	100DM	0
Economie 1992:	100DM	0
Economie 1993:	100DM	300DM
Timpul de recuperare	3 ani	3 ani

Ambele proiecte necesită o investiţie de 300 de mărci şi 3 ani pentru a recupera aceşti bani. Analiza recuperării fondurilor arată că aceste două proiecte au aceeaşi valoare. Aceasta este însă o concluzie pripită. În cazul proiectului A banii se reîntorc mai repede decît în cazul proiectului B, şi prin urmare ei pot fi reinvestiţi. Din punctul de vedere contabil prima variantă este mai bună.

Calculul reîntoarcerii fondurilor cu siguranţă nu este metoda care trebuie folosită în cazul proiectelor mari şi de durată. Există însă proiecte simple, care au o perioadă de amortizare atît de scurtă, încît e superfluă analiza valorii actuale. La unele LAN-uri se ajunge la o perioadă de amortizare de doar cîteva luni, folosirea lor ducînd la o creştere foarte mare a productivităţii şi la economii mari.

Dacă perioada de amortizare este mai mică de un an, atunci de regulă nu este necesară analiza detaliată a costurilor. Ea nu mai aduce informaţii importante, suplimentare.

Costuri de exploatare

Alternativa 1: LAN

An	Descriere	Costuri	Inflaţie în %	Costuri inclusiv inflaţie
1991	Şcolarizare	10000	5	10000
	Întreţinere	4000	5	4000
	Productivitate	0	5	0
1992	Şcolarizare	5000	5	5250
	Întreţinere	4000	5	4200
	Productivitate	0	5	0
1993	Şcolarizare	0	5	0
	Întreţinere	4000	5	4410
	Productivitate	0	5	0
1994	Şcolarizare	0	5	0
	Întreţinere	4000	5	4631
	Productivitate	0	5	0
1995	Şcolarizare	0	5	0
	Întreţinere	4000	5	4862
	Productivitate	0	5	0
1996	Şcolarizare	0	5	0
	Întreţinere	4000	5	5105
	Productivitate	0	5	0

Alternativa 2: Mainframe

An	Descriere	Costuri	Inflaţie în %	Costuri inclusiv inflaţie
1991	Şcolarizare	0	5	0
	Întreţinere	8000	5	8000
	Productivitate	20000	5	20000
1992	Şcolarizare	0	5	0
	Întreţinere	8000	5	8400
	Productivitate	30000	5	31500
1993	Şcolarizare	0	5	0
	Întreţinere	8000	5	8820
	Productivitate	30000	5	33075
1994	Şcolarizare	0	5	0
	Întreţinere	8000	5	9261
	Productivitate	30000	5	34729
1995	Şcolarizare	0	5	0
	Întreţinere	8000	5	9724
	Productivitate	30000	5	36465
1996	Şcolarizare	0	5	0
	Întreţinere	8000	5	10210
	Productivitate	30000	5	38288

În primele două tabele se arată costurile de exploatare în ambele variante.Dacă se menţine soluţia mainframe vor apărea pierderi de productivitate.

Reducerea costurilor prin "downsizing"

Procedeul de "downsizing" este acum la modă. Într-un proiect tipic de downsizing trebuie analizate costurile pentru transferarea exploatării unor aplicaţii de pe mainframe pe calculatoare mai ieftine, comparativ cu costurile menţinerii modului actual de exploatare a aplicaţiilor.

Trebuie calculate costurile şi veniturile în ambele alternative. Pe de o parte este vorba de valori care sînt evidente şi care apar o singură dată sau de mai multe ori. Pe de altă parte există costuri şi venituri, care nu pot fi apreciate la prima vedere. Ele apar de mai multe ori.

Primul pas al analizei constă în estimarea valorilor costurilor şi veniturilor. Pentru evaluarea financiară sînt necesare atît pricepere managerială cît şi putere de imaginaţie.

Sînt mult mai mulţi cei pe care mai degrabă această evaluare, şi nu formulele matematice, îi duc la disperare. Dacă se uită un factor relevant de costuri sau anumite economisiri, întreaga analiză este incorectă, chiar dacă toate datele sînt confirmate prin zeci de pagini de documentaţie. Printr-o astfel de greşeală analiza, şi prin ea proiectul, pot fi puse sub semnul întrebării. La acest pas ar trebui să existe disponibil timp suficient, care să permită şi consultarea altor colaboratori, obiectivi.

Printre costurile care apar o singură dată sînt şi următoarele:

- dezvoltarea aplicaţiilor

Durata de reîntoarcere a capitalului

Alternativa 1: Aplicaţia se rulează pe un LAN

An	Costuri de exploatare	Costuri de exploatare după impozitare	Amortizare	Amortizare după impozitare *	Investiţie	Flux de capital după impozitare
1991	14000	9240	10000	3400	50000	55840
1992	9450	6237	16000	5440		797
1993	4410	2911	9600	3264		-353
1994	4631	3056	5750	1955		1101
1995	4862	3209	5750	1955		1254
1996	5105	3369	2900	986		2383
Total	42458	28022	50000	17000	50000	61022

Alternativa 2: Aplicaţia rămîne pe mainframe

An	Costuri de exploatare	Costuri de exploatare după impozitare	Amortizare	Amortizare după impozitare *	Investiţie	Flux de capital după impozitare
1991	28000	18480	0	0	0	18480
1992	39900	26334	0	0	0	26334
1993	41895	27651	0	0	0	27651
1994	43990	29033	0	0	0	29033
1995	46189	30485	0	0	0	30485
1996	48499	32009	0	0	0	32009
Total	248473	163992	0	0	0	163992

* Pentru calculul din exemplu s-a presupus un procent de impozitare de 30%.

Acest calcul determină perioda de reîntoarcere a capitalului. Şi din acest exemplu se poate observa că un LAN este mai avantajos.

- hardware
- software pentru sistemul de operare
- cursuri de iniţiere
- realizarea conexiunilor iniţiale
- restructurarea platformei reţelei
- instalarea etc.

Printre costurile şi economisirile palpabile, şi care se repetă, sînt şi:

- măsuri de perfecţionare pentru utilizatori şi manageri
- întreţinerea hard-ului şi soft-ului
- asistenţa aplicaţiilor
- modificări de cablaj
- administraţie locală şi globală, la nivelul întregii întreprinderi
- siguranţă
- suport
- distribuţia soft-ului
- telecomunicaţii
- productivitate etc.

Anumite costuri şi economisiri nu sînt direct măsurabile, dar ele există totuşi:

- flexibilitatea întreprinderii
- mulţumirea colaboratorilor
- "imagine" mai dinamică a întreprinderii
- limite de timp mai puţin strînse pentru cei care au de luat decizii
- ergonomia locului de muncă
- administrarea reţelei (şi la nivelul secţiilor) etc.

Indiferent cît de mult ar fi detaliată o analiză, ea este doar atît de bună ca şi cifrele pe care le conţine. Nu trebuie uitate costurile pentru colaborări. La multe proiecte acesta este cel mai important factor de costuri. De regulă, colaboratorii costă mult mai mult decît hard-ul şi soft-ul. De aceea costurile cu personalul nu trebuie evaluate prea jos.

Deoarece costurile "invizibile" sînt foarte greu de calculat, de multe ori se preferă soluţia, ale cărei costuri sînt mai transparente. Dar, deseori, această soluţie nu este alternativa optimă pentru întreprindere. Ar trebui măcar încercată estimarea rezonabilă a valorii costurilor invizibile.

Conducătorii unei întreprinderi acceptă mai uşor un proiect dacă este însoţit de cifre credibile.

Directorii şi contabilii ezită de obicei la cumpărarea unor noi calculatoare, deoarece nu li se pare o investiţie necesară. Totuşi, costul unui PC este doar cît salariul pentru cîteva zile al unui manager. Dacă în ce priveşte costurile şi economisirile "invizibile" se insistă pe creşterea probabilă a productivităţii, atunci lupta este, de regulă, deja pe jumătate cîştigată.

Analiza costurilor pentru downsizing

Sînt puse faţă în faţă ambele alternative, atît soluţia LAN cît şi soluţia mainframe. Costurile se calculează pe baza unor factori de productivitate, care includ costurile colaborărilor, inclusiv salarii şi impozite.

Analiza din exemplu este făcută pentru o perioadă de şase ani. În această perioadă hard-ul de LAN se amortizează complet, de multe ori,

Valoarea actuală netă

Alternativa 1: Aplicaţia se rulează pe un LAN

An	Flux de capital după impozitare	Taxă de scont de 10%
1991	55840	55840
1992	797	725
1993	-353	-292
1994	1101	828
1995	1254	856
1996	2383	1480
Total	61022	59436

Valoarea actuală netă = 59436

Alternativa 2: Aplicaţia rămîne pe mainframe

An	Flux de capital după impozitare	Taxă de scont de 10%
1991	18480	18480
1992	26334	23940
1993	27651	22852
1994	29033	21813
1995	30485	20821
1996	32009	19875
Total	163992	127782

Valoarea actuală netă =127782

Acest calcul corelează fluxul de capital după impozitare cu valoarea în timp a banilor.

amortizarea este terminată după trei ani. Costurile cu şcolarizarea, cu întreţinerea hard-ului şi soft-ului au fost calculate pentru fiecare an în parte. Se porneşte de la ideea că rata de inflaţie se ridică la 5% anual. Valoarea lui este inclusă, în exemplu, în costurile de exploatare. Acest lucru este necesar deoarece analiza se bazează pe valoarea actuală a mărcii. Ca şi în toate celelalte domenii, costurile vor creşte în următorii ani, şi de acest lucru trebuie să se ţină seama în analiza costurilor.

Deşi deseori utilizate, aceste tabele nu iau în considerare costurile care se pot economisi prin alte soluţii. Ele permit doar o comparaţie a costurilor dintr-un interval de timp. Se pleacă de la premisa că una din alternative este obligatorie (de exemplu facturarea. Este evident că întreprinderea are nevoie pentru aceasta de o soluţie PAD). Întrebarea este doar: care dintre alternative este cea mai rentabilă?

Stabilirea costurilor de exploatare este doar un pas din analiza integrală a costurilor. Cu această ocazie, se adună informaţii care vor reintra mai tîrziu în analiză. Cu analiza costurilor de exploatare se obţine o primă impresie asupra rentabilităţii proiectului, dar ea nu reprezintă un instrument de analiză de sine stătător.

După ce au fost determinate costurile pentru perioada de analizat atît pentru soluţia LAN cît şi pentru varianta mainframe, ele trebuie evaluate prin intermediul unui program de calcul tabelar. Pentru proiectul dat ca exemplu trebuie stabilite valoarea actuală netă şi perioada de reîntoarcere a capitalului. Valorile de bază ale acestui spreadsheet provin din calculul costurilor de exploatare.

Aici apare pentru prima oară amortizarea. Valoarea finală din coloana

Din vocabularul financiar

La fel ca şi tehnicienii, şi experţii în finanţe folosesc propria lor terminologie. Cine doreşte să se înţeleagă cu ei, trebuie să le cunoacă "limba". Explicăm în continuare cîţiva termeni, folosiţi foarte des:

Fluxul de capital

Scopul cel mai important al unei analize a costurilor este de a constata cît de mult influenţează un anumit proiect finanţele unei intreprinderi. Toate proiectele cer mai întîi nişte investiţii, se cheltuie deci nişte bani. După realizarea lor se aşteaptă ca banii să reintre în întreprindere, sub forma unor reduceri ale costurilor. Trebuie arătat, pentru fiecare interval de timp din plan cîţi bani curg şi în ce direcţie. Aceste mişcări de bani se denumesc flux de capital.

Perioada de analiză

Aici se determină perioada în care proiectul va influenţa fluxul de capital al întreprinderii. Trebuie luate în seamă ciclurile de viaţă ale resurselor necesare: de regulă, pentru soft 3 ani, pentru hard 5 ani şi pentru cablări 10-15 ani.

Este foarte dificilă o prognozare a fluxului de capital pe o perioadă mai mare de cinci ani, deoarece pot apărea mulţi factori neaşteptaţi. Acest lucru este valabil mai ales la proiecte care se bazează pe calculatoare. Are sens alegerea unui interval de timp cuprins între 3 şi 5 ani.

Valoarea în timp a banilor sau bani numerar şi discont

Atunci cînd se împrumută 100 de mărci nimănui nu-i este indiferent, dacă primeşte banii mîine înapoi sau abia peste un an. Dacă banii lipsesc un an, ei nu pot fi cheltuiţi sau nu pot fi depuşi la bancă ca să aducă o anumită dobîndă. Valoarea în timp a banilor se bazează pe principiul că "banii în mînă" valorează mai mult, decît banii care se vor obţine în viitor.

Dacă, de exemplu costurile unui proiect însumează 10 mărci, şi se aşteaptă economisiri de 5 mărci în fiecare din următorii 2 ani, cele două valori se compensează. Din punctul de vedere al experţilor financiari însă, în acest caz este vorba despre un minus. Acest factor se numeşte discont. Cu ajutorul lui se calculează valoarea în numerar a banilor la o dată ulterioară, în exemplul nostru fiind vorba de un an, respectiv de 2 ani.

Costurile de capital şi dobînda

Noţiunile rată de discont şi costuri de capital sînt interschimbabile. Costurile de capital se calculează din modul în care se finanţează o întreprindere. De obicei, acesta este o combinaţie între acţiuni, venituri şi cheltuieli pentru bani împrumutaţi.

Acest calcul nu trebuie făcut de către administratorul analizei costurilor. El poate afla rata corespunzătoare de discont de la contabilitate.

Dobînda este în principiu un procentaj la împrumut. Ea se referă la reîntoarcerea capitalului din investiţia iniţială şi la economisiri datorate proiectului respectiv în perioada de analiză.

Impozite

Statul obţine cca. 50% din veniturile întreprinderii. Astfel, fiscul devine "partenerul" cel mai important al întreprinderii. Este însă posibilă scăderea din impozite a costurilor unui proiect aducător de venituri ca şi a cheltuielilor întreprinderii. Este şi acesta un factor important al analizei costurilor.

Amortizarea

Se investeşte atît în calculatoare cît şi în alte resurse ale întreprinderii. Un 486, care rezolvă o anumită problemă, nu va rămîne pe veci în exploatare. Odată şi o odată tot vine rîndul unei noi investiţii.

Legea permite scăderea costurilor din impozit pe o anumită perioadă de timp. Se presupune că aceste aparate sînt "consumabile" şi în plus duc pe termen lung la venituri crescute, impozabile. Costurile de amortizare reduc însă pe moment venitul total impozabil. Cu cît amortizarea este mai mare, cu atît impozitele ce sînt de plătit sînt mai mici.

De cele mai multe ori calculul amortizării este foarte uşor de efectuat. Programele de calcul tabelar conţin de regulă funcţii corespunzătoare, care calculează automat amortizarea corectă. Şi în acest caz contabilul devine partener: el poate explica cum este aplicată amortizarea de către întreprindere.

"fluxul de capital după impozitare" se calculează din costurile de exploatare după impozitare la care se adaugă investiţiile în calculatoare şi se scade amortizarea după impozitare. Dacă se foloseşte pentru asemenea analize un program de calcul tabelar uzual cum ar fi "Lotus 1-2-3", calculul se face automat. Doar amortizarea şi investiţia trebuie introduse.

Valoarea actuală, reîntoarcerea capitalului şi dobînzile

Pasul următor este calcularea valorii actuale. Şi în acest caz, programele de calcul tabelar dispun de funcţii proprii. Aceste formule pot fi elaborate şi individual, dar este mult mai complicat, şi în cele mai multe cazuri, se va evita acest lucru. Urmează calculul VAN (valorii actuale nete), care se obţine scăzînd taxa de scont (în exemplu ea este de 10%).

Pentru exemplul de mai sus s-a calculat VAN de 59.436 mărci. VAN pentru soluţia mainframe este de 127.782 mărci. Deci alternativa LAN este cu 68.346 mărci mai ieftină. În mod concret rezultă de aici că proiectul preconizat este mult mai rentabil dacă funcţionează pe LAN decît pe un calculator mare, cu toate că acesta există deja în întreprindere.

În acest punct, analiza ar fi de fapt terminată. Calculaţia valorii actuală este atît de clară încît aduce destule argumente pentru proiect. Totuşi uneori se doreşte şi efectuarea unei analize a reîntoarcerii capitalului. Se pune întrebarea: cît timp va dura pînă cînd investiţia va intra din nou în intreprindere? Intrarea capitalului rezultă din calculul economisirilor în cadrul costurilor generale ale unui LAN faţă de cele ale unui mainframe.

Acest calcul este foarte simplu.Trebuie doar scăzute costurile anuale pentru LAN din cele corespunzătoare pentru mainframe. În exemplul nostru se vede că în primul an diferenţa de 37.360 mărci este în favoarea mainframe-ului. Motivul este investiţia de cumpărare a LAN-ului, care este de 50.000 de mărci. În următorii ani balanţa se înclină în favoarea LAN reţeaua devenind mult mai avantajoasă decît mainframeul. În acest caz, capitalul se reîntoarce în întregime după doi ani, iar economiile aduse în următorii ani aduc argumente inatacabile în favoarea trecerii la o reţea locală.

În unele cazuri, în locul sau pe lîngă calculul VAN se calculează "dobînda". Trebuie găsită o dobîndă limită care duce la o valoare nulă de capital, deci numerarul de venituri şi cheltuieli să fie la fel de mare. Această dobîndă procentuală internă se determină cu o formulă cunoscută în contabilitate

Dobînda limită

An	Flux de capital LAN după impozitare	Flux de capital mainframe după impozitare	Diferenţa	Taxa de scont 69%
1991	55840	18480	-37360	-37360
1992	797	26334	25537	15292
1993	-353	27651	28004	10041
1994	1101	29033	27932	5997
1995	1254	30485	29231	3758
1996	2383	32009	29626	2281
Total	61022	163992	102970	9

Dobînda limită corectă se calculează cu o formulă. Această valoare oferă informaţii despre taxa de scont, în cazul nostru ea fiind de 67%

. Astfel se obţine plata efectivă a dobînzilor unui proiect de investiţie înainte de scăderea plăţii dobînzilor. În proiectul din exemplu valoarea finală a acestui calcul este de 9 mărci şi deci foarte aproape de valoarea ideală 0. Această valoare a fost atinsă prin aplicarea a diferite dobînzi limită. Ce ne spune o dobîndă limită de 67%? Dacă întreprinderea trebuie să împrumute capitalul necesar de la bancă la o dobîndă de 15%, înseamnă că dobînda limită minimă trebuie să fie de cel puţin 15%. Deoarece ea a fost stabilită la 67% înseamnă că proiectul va fi cu siguranţă acceptat.

Proiectele care sînt calculate la o dobîndă limită ridicată sînt puse la îndoială de contabili. Ei sînt înclinaţi să verifice în amănunt aceste cifre. De aceea, analizele care prezintă dobînzi limită ridicate, nu vor fi prezentate în această formă.

Documentarea succesului

Analiza costurilor a convins atît conducerea cît şi contabilii întreprinderii. Proiectul a fost realizat şi s-a dovedit eficace şi productiv. Nu trebuie uitată acum documentarea succesului, cu toate economiile şi avantajele lui. Mai ales proiectele "downsizing" provoacă nelinişte, deoarece pentru întreprindere ele înseamnă mari modificări organizatorice.

Chiar dacă proiectul întrece toate aşteptările, vor exista mereu persoane care nu se vor putea acomoda cu noile structuri organizatorice.Pentru a veni în întîmpinarea argumentelor lor, este utilă elaborarea unei documentaţii cuprinzătoare.

Este indicat ca de la bun început contabilitatea să fie implicată în proiect. Oricum la început sînt necesare informaţii în legătură cu amortizarea, impozitele şi fluxul capitalului; informaţii ce pot fi obţinute numai de la aceşti specialişti.

În afară de aceasta, se poate pleca de la premisa că aceştia vor efectua o analiză de control a proiectului după ce acesta a fost pus în practică. Ei sînt cu siguranţă mai maleabili dacă sînt implicaţi de la bun început şi dacă trebuie să controleze propriile lor cifre.

Chiar dacă, în mod normal, contabilitatea va prelua cea mai mare parte a muncii pentru analiza costurilor, este important să se cunoască măcar în mare procedeul. Astfel se pot înţelege cifrele şi motivele care determină o anumită decizie. Dacă analiza este făcută independent şi li se prezintă experţilor financiari în "limba" lor, se poate conta pe acceptarea proiectului.

(I.M)

Servere de baze de date

În cele ce urmează vom încerca o trecere în revistă a SGBD-urilor care implementează modelul client-server în reţele locale de calculatoare (Local Area Network). Produsele care urmează a fi prezentate fac parte din categoria serverelor de baze de date (database server) şi sînt proiectate pentru a rula pe un LAN-server uzual, înzestrat cu un sistem de operare în reţea solid şi pentru a furniza servicii de gestiune a datelor staţiilor de lucru-client.

Serverele de baze de date incluse în reţele locale de calculatoare au început să preia sarcini odinioară îndeplinite numai de SGBD-uri operînd pe minicalculatoare sau sisteme mari, inclusiv aplicaţii on-line de prelucrare a tranzacţiilor. De asemenea, serverele de baze de date permit accedarea partajată a informaţiei, sigură şi simultană, unei largi varietăţi de aplicaţii utilizate în mod curent şi posedînd o interfaţă-utilizator prietenoasă cum sînt, de pildă, aplicaţiile de calcul tabelar sau editoarele de texte.

Softul de server de baze de date destinat microcalculatoarelor rulează - după cum lesne se poate închipui - pe serverul reţelei. Cînd o aplicaţie-client are nevoie de regăsirea sau actualizarea unei informaţii formulează o solicitare, de obicei în SQL (Structured Query Language), către serverul de baze de date. Acesta prelucrează cererea, regăseşte şi actualizează înregistrările şi, fie că le returnează pe acestea, fie că un cod de stare este returnat aplicaţiei.

În ceea ce priveşte mai răspînditul model al serverului de fişiere (file server), se poate spune că acesta foloseşte serverul reţelei ca pe o unitate de hard disk şi nimic mai mult. În consecinţă, fişiere întregi uneori sînt transmise prin reţea pentru a fi prelucrate de staţia de lucru-client. Cu alte cuvinte, transferul de date care are loc, într-un SGBD monoutilizator, între unitatea de disc fix şi UC, pe un bus relativ larg, trebuie să se desfăşoare, în modelul serverului de fişiere, pe mediul de comunicare al reţelei, evident mai strîmt şi, în plus, folosit simultan şi de alţi clienţi din reţea. Modelul client-server evită acest neajuns.

Însă serverele de baze de date pot face mai mult decît să descongestioneze mediul de comunicare al reţelei. Caracteristici specifice le permit să asigure integritatea datelor (în unele cazuri, chiar integritatea referenţială a acestora), să sprijine gestiunea tranzacţiilor şi să recupereze toate tranzacţiile în situaţia unei defecţiuni a sistemului. Unele servere de baze de date prezintă extensii către limbajul SQL care le oferă posibilitatea de a impune la nivel central anumite condiţii sau reguli cum ar fi, de pildă: "nu accepta o comandă de la un client insolvabil" sau "adaugă la valoarea mărfii 5%, reprezentînd ICM-ul".

Nu este necesară implementarea acestor reguli în fiecare aplicaţie care accedează datele, fiind suficientă o singură implementare. Dacă o regulă se schimbă, de exemplu dacă ICM-ul creşte la 10% atunci ne putem consola cu faptul că modificarea corespunzătoare trebuie operată într-un singur loc. Calculatoarele-client care vin în contact cu o arhitectură client-server prin intermediul unui soft front-end au avantajul că nu trebuie să îndeplinească toate funcţiile de prelucrare a datelor, o parte dintre acestea revenind serverului de baze de date. Astfel, calculatoare- client mai puţin performante se comportă mai bine decît ar putea să o facă într-un model "file server". Şi un alt avantaj major al arhitecturilor client-server: problema creşterii performanţelor întregului sistem se reduce la creşterea performanţelor calculatorului pe care lucrează serverul de baze de date în timp ce o operaţie similară asupra unui server de fişiere nu reprezintă o soluţie, atît timp cît ea nu se extinde şi asupra staţiilor de lucru.

Înainte de a trece la prezentarea succintă a cîtorva dintre cele mai cunoscute servere de baze de date, subliniem rolul important pe care îl au în arhitecturile client-server instrumentele front-end, de legătură între serverul de baze de date (back-end) şi utilizator. Cîteva dintre cele mai răspîndite asemenea instrumente sînt: Lotus 1-2-3, Paradox, dBASE IV, DataEase, ObjectVision, R&R Report Writer, SQL Windows ş.a. În final, două menţiuni: între paranteze este indicat producătorul iar preţurile sînt informative.

Emerald Bay 2.27 (RSPI)

Emerald Bay este un SGBD multiutilizator, destinat în special programatorilor, care oferă o arhitectură client-server pentru PC-uri. Vulcan, o adaptare a limbajului Xbase pentru Emerald Bay, oferă un interpretor, un compilator EXE şi facilităţi care privesc funcţii definite de utilizator, operarea cu ajutorul mouse-ului, salvarea şi restaurarea imaginilor ecranului şi tablouri multidimensionale. C Developer's Toolkit, o bibliotecă de funcţii C care accedează bazele de date Emerald Bay dintr-un program C, conţine biblioteci de imagini ecran care permit includerea formatelor de ecran generate cu Forms Design Utility în programe scrise în C. Şi pentru Pascal a fost prevăzută o "trusă", Pascal Developer's Toolkit cu ajutorul căreia se pot accesa baze de date EB din programe Pascal. Fiecare produs sau "trusă" Emerald Bay include un Engine pentru prelucrarea bazelor de date EB din programe scrise în C sau Pascal, un Database Administrator, funcţii de import/export (Rosetta), editoare de machete de ecran (Forms Designer/Screen Painter), un editor de rapoarte (Report Writer), unul de texte (Text Editor) şi o versiune minimală, pentru 2 utilizatori, a serverului. Vulcan şi C Developer's Toolkit costă cîte 395$ iar Pascal Developer's Toolkit, 199$. Preţul serverului EB este de 199$ pentru 4 utilizatori, 399$ pentru 8 şi de 599$ pentru un număr nelimitat de utilizatori.

Ingres for OS/2, DOS (Ingres)

Ingres Server for OS/2 şi Ingres Tools for DOS sînt instrumente care realizează conectarea PC-urilor-client la SGBD-ul relaţional Ingres rulînd pe un server sub UNIX sau VAX/VMS, prin intermediul sistemelor de operare în reţea LAN Manager (Microsoft), NetWare (Novell) sau 3+ Open (3Com). Acestea lucrează pe PC-uri unde, după ce traduc cererea în limbaj SQL, o transmit prin reţea către server. Acesta optimizează interogarea şi efectuează gestiunea datelor, returnînd sau memorînd informaţia corespunzătoare. Ingres Server for OS/2 se vinde la preţul de 1.995$.

Microsoft SQL Server 1.1 (Microsoft)

Microsoft SQL Server este un sistem de gestiune a bazelor de date relaţionale (SGBDR) pentru reţele de PC-uri care oferă o gestiune centralizată a datelor, cu accent pe securitatea şi integritatea acestora precum şi pe controlul utilizatorilor. Este sprijinit de o gamă largă de aplicaţii front-end. SQL Server, împreună cu LAN Manager şi DCA Microsoft Comm Server alcătuiesc o familie de produse Microsoft proiectată să rezolve orice problemă care poate fi ridicată de funcţionarea unei reţele de calculatoare. Probabil că în funcţie de numărul de probleme rezolvate variază şi preţul: între 995 şi 3995$.

NetWare SQL 2.11 (Novell)

NetWare SQL 386 este implementat ca un modul încărcabil NetWare (Net-

Ware Loadable Module) ceea ce înseamnă că este puternic integrat în sistemul de operare în rețea NetWare 386, spre deosebire de un VAP (Value Added Process) care rulează "deasupra" sistemului de operare, fără să aibă vreo interferență cu acesta. Acest fapt îl face mai robust și mai performant. Poate executa simultan prelucrări vizînd mai multe baze de date. Aplicațiile front-end pot acceda ambele versiuni ale sistemului, 286 și 386, fără nici o modificare. NetWare SQL folosește protocoalele NetWare uzuale (Internetwork Packet Exchange și Sequenced Packet Exchange) pentru a acceda resursele serverului. Funcțiile de securitate și de administrare a bazei de date sînt centralizate la nivel de server. Cu toate că NetWare SQL respectă prevederile referitoare la comenzi SQL, conținute în standardul ANSI Level 1, oferă și funcții suplimentare.

Oracle Server 6.0 (Oracle Corp.)

Oracle Server permite rularea de aplicații multiutilizator în arhitecturi client-server sau pe minisisteme și utilizarea simultană a aceleiași baze de date. Utilizatorii pot accesa datele direct de la stații de lucru DOS, OS/2, UNIX sau Macintosh. Poate folosi atît protocolul SPX/IPX pentru conectarea rețelelor funcționînd sub Novell NetWare cît și protocolul TCP/IP pentru UNIX, precum și protocoale asincrone. Realizează prelucrarea tranzacțiilor și recuperarea automată a sistemului. Versiuni pentru OS/2, SCO UNIX, XENIX și AT&T V/386. Preț: 51.00$ pentru 8 utilizatori în configurații 386/486.

SQLBase Server 4.1 (Gupta Technologies)

SQLBase este un server de baze de date care furnizează o implementare completă de SQL rulînd atît sub DOS cît și sub OS/2. Ambele versiuni transformă un PC/AT într-un server de baze de date care poate fi accedat de mai multe PC-uri integrate într-o rețea compatibilă NetBios. De asemenea, ambele includ o interfață de programare în C a aplicațiilor pentru Microsoft Windows și sînt în curs de elaborare instrumente pentru OS/2 Presentation Manager. Setul de dezvoltare multiutilizator costă 1995$/server iar versiunea monoutilizator, 955$. Instrumentul front-end SQLWindows, produs tot de Gupta Technologies, conține un editor grafic de formate ecran tip Microsoft Windows. Limbajul procedural intern SQLWindows, numit SAL, poate include instrucțiuni SQL imbricate pentru accedarea bazei de date, poate utiliza obiecte create cu editorul de formate și este sprijinit de un editor intern SQLWindows. SQLNetwork, al aceleiași firme, permite conexiuni transparente între aplicațiile SQLBase și alte baze de date SQL existente pe minicalculatoare sau sisteme mari.

(R.G.)

Visual Basic - programare intuitivă

(continuare din pag. 24)

Între aceste 2 linii trebuie inserate instrucțiunile ce vor trebui executate atunci cînd se tastează pe mouse. Un lucru e clar: Visual Basic este, spre deosebire de alte produse ca Object Vision al firmei Borland sau Toolbook al firmei Asymetrix, gîndit, înainte de toate, ca un soft pentru că, în orice caz trebuie făcute programe. Ceea ce înseamnă că, în ciuda existenței unui "Help" foarte bun, cu multe exemple, este absolut necesară cunoașterea instrucțiunilor, lucru ce nu poate fi cerut unui profan.

În ajutorul programatorilor

Fereastra de instrucțiuni, (de cod), arată ca un editor de texte. Pe marginea superioară sînt 2 rubrici. Cea din stînga conține o listă a controalelor formularului activ. Din această listă se poate alege cărui control să i se asocieze mecanismul de desfășurare al unui eveniment. În cea din dreapta apar procedurile care, pentru controlul ales, pot fi sau sînt deja programate. Cînd nu este absolut necesar să se cuprindă toate evenimentele în program, evenimentele codificate deja vor fi vizualizate.

Astfel, Visual Basic, spre deosebire de Windows, îi economisește programatorului o mulțime de timp: el nu trebuie să se mai ocupe de mulțimea de mesaje pe care nucleul Windows le trimite programului activ, și el se poate concentra asupra interacțiunii cu utilizatorul, restul rezolvă Visual Basic în locul lui.

La introducerea liniilor de program, Visual Basic veghează asupra corectitudinii sintaxei. Astfel, greșelile de tastare sînt recunoscute imediat și pot fi corectate. Acest mod de lucru este cunoscut deja din Quick-Basic și se bazează pe transformarea codului sursă într-un cod intermediar (Threaded-P-Code). Acesta nu corespunde formei finale a programului, dar poate fi oricînd lansat din Visual Basic. Execuția codului P nu este așa de rapidă ca cea a codului final, dar programul poate fi oricînd testat sub această formă și eventualele greșeli pot fi eliminate foarte repede fără a mai pierde timp cu compilarea.

Pentru depanarea programelor, Visual Basic oferă tot ajutorul ce se așteaptă de la un limbaj modern de programare. Astfel, pot fi fixate puncte de întrerupere (Breakpoint) care provoacă întreruperea programului și permit vizualizarea și chiar modificarea conținutului unor variabile. Programul poate fi modificat și în timpul unei întreruperi și apoi se poate continua rularea lui. Deci, subliniem, rularea programului nu trebuie reluată de la început în cazul efectuării unor modificări. După ce programul a fost testat "la sînge" el poate fi compilat cu compilatorul Visual Basic-ului rezultatul fiind un fișier EXE. Programul executabil astfel obținut poate fi rulat independent de Visual Basic și poate fi vîndut sau dat fără taxă de licență.

Dezvoltări utile în perspectivă

Prin apariția Visual Basic-ului, Microsoftul a făcut un salt în viitor. Odată cu apariția acestui pachet programarea aplicațiilor Windows nu se mai limitează la cercul programatorilor ce sînt familiarizați cu Windows - API. Întrucît Visual Basic permite și schimbul dinamic de date (DDE), el devine o platformă potrivită pentru integrarea aplicațiilor existente. Nu trebuie renunțat sub nici o formă la rutinele C sau asamblor existente; ele pot fi folosite prin intermediul DLL-ului (Dynamic Link Libraries). Premisa reușitei este bineînțeles un compilator corespunzător, altfel programele existente nu pot fi rulate sub Visual Basic.

În curînd Microsoft va scoate și un Control Development Kit care va face posibilă definirea proprie a unor controale folosibile apoi în formulare. Visual Basic-ul funcționează pe orice calculator pe care poate funcționa Windows-ul. Totuși, se recomandă următoarea configurație minimă: un calculator 386SX (16MHz) cu o memorie internă de 2 MBytes, un hard-disk de 40 MBytes, un monitor VGA și un mouse. Prețul acestui pachet este de 600 DM.

(Radovici Vera)

Sistemul de gestiune al fişierelor MS-DOS

Atît în cazul intrărilor, fie ele de la tastatură sau de la mouse, în cazul ieşirilor, fie la imprimantă sau pe ecran, cît şi în cazul multor altor operaţii, programatorii se comportă de multe ori ca şi cum nu ar exista MS-DOS. Ei folosesc BIOS-ul sau adresează direct hardware-ul. Un singur domeniu al programării rămîne întotdeauna în atenţia sistemului de operare: manipularea fişierelor şi a directoarelor se face doar prin intermediul funcţiilor MS-DOS. Există motive întemeiate în acest sens.

Fişierele şi directoarele formează partea vizibilă a unei componente complexe şi foarte importante a MS-DOS şi anume sistemul de gestiune al fişierelor (S.G.F.). Acesta este cel care dă dischetelor şi harddisk-urilor structura lor şi face posibil accesul foarte simplu al utilizatorilor de PC-uri la conţinutul acestor purtători de date. Dacă vă gîndiţi care sînt acţiunile care se repetă zilnic şi care depind de accesul la fişiere, atunci veţi ajunge la concluzia că sistemul de operare MS-DOS nu s-ar putea închipui fără S.G.F.

Închipuiţi-vă de exemplu, ce se întîmplă cînd apelaţi un program. În primul rînd MS-DOS va căuta programul (el citeşte directoarele căutînd în conţinutul numele programului). Dacă l-a găsit, atunci îl va încărca (se citeşte fişierul program şi conţinutul va fi copiat în memoria de lucru). În fine, MS-DOS lansează programul, care în cele mai multe cazuri execută din nou accedarea asupra fişierelor, pentru a citi datele necesare funcţionării sale. Fiecare dintre aceste accedări indiferent că ele sînt lansate de MS-DOS sau de alt program, sînt derulate de S.G.F. Vedeţi deci că fără S.G.F. numic nu merge.

S.G.F. stă ca bază pentru structura fizică a dischetelor sau a harddisk-urilor. Deoarece aceste două tipuri de memorii de masă sînt structurate aproape identic, în continuare vom utiliza noţiunea *disc*, atunci cînd deosebirile dintre dischetă şi harddisk sînt nesemnificative. Structura fizică a acestor medii de memorare este cît se poate de simplă. Întregul disc este divizat în domenii de aceeaşi dimensiune, denumite *sectoare*. Sectoarele au aceeaşi dimensiune şi se deosebesc între ele doar prin poziţia fizică de pe disc. Din această cauză ele mai sînt denumite şi *sectoare fizice*. Un sector poate fi identificat în două moduri. Controller-ul harddisk-ului utilizează adresa fizică a fiecărui sector, în timp ce MS-DOS procedează mai simplu: sectoarele sînt numerotate crescător începînd de la 0. La această numerotare se face referire atunci cînd este vorba despre *numărul sectorului*.

Structura prezentată este produsă în urma procesului de *formatare*. Din păcate, în cazul sistemului de operare MS-DOS, această noţiune are două sensuri, şi aceasta din cauza unor motive istorice: atunci cînd a fost creat sistemul de operare MS-DOS, PC-urile nu dispuneau decît de lectoare de dischete. Pentru ca o dischetă să poată fi utilizată pentru scopul pentru care era destinată, deci ca suport de date, a fost creată comanda *format*, care rezolva două probleme. Pe de o parte această comandă realiza împărţirea dischetei în sectoare, iar pe de altă parte, comanda *format* crea şi structura logică de date de care avea nevoie MS-DOS pentru a putea memora date pe dischetă. Şi în mod natural aceste două ope-

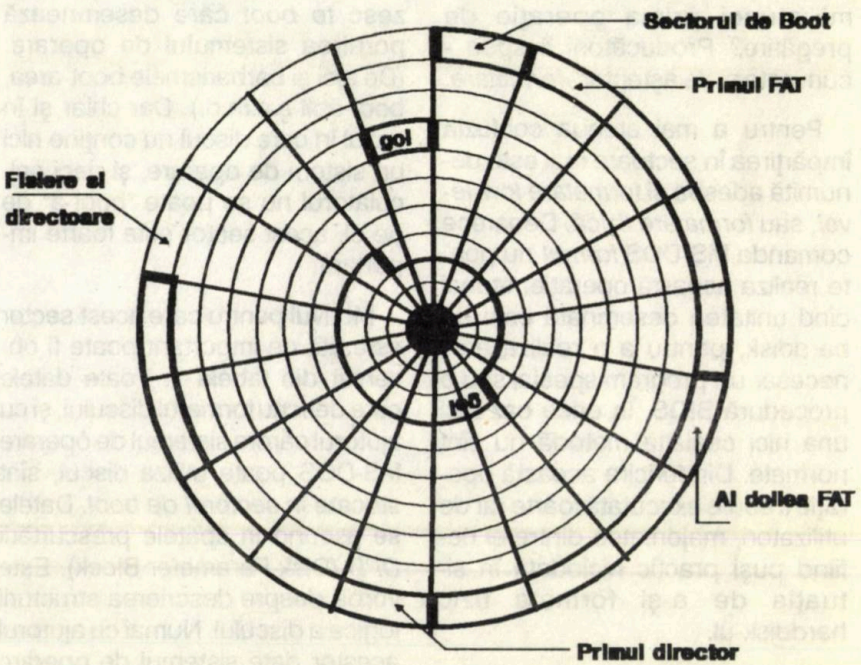


Fig. 1: Vedere de ansamblu asupra structurii de date create de sistemul de operare MS-DOS în urma formatării pe un disc

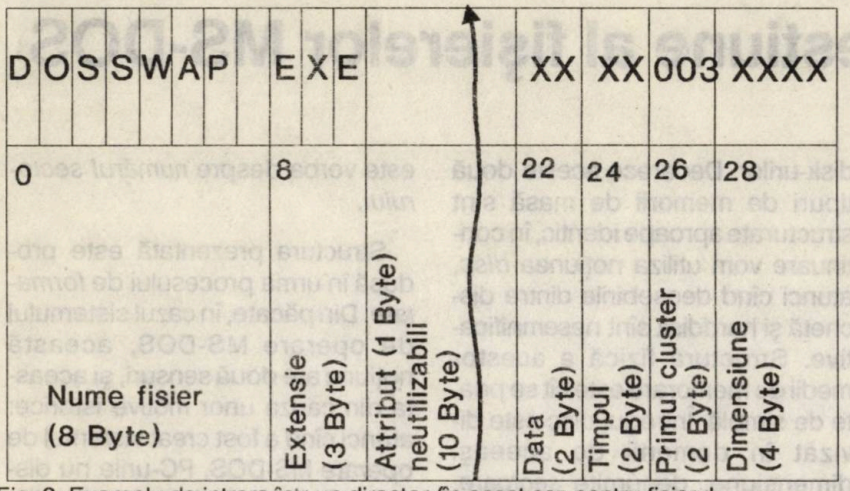


Fig. 2: Exemplu de intrare într-un director (în acest caz pentru fişierul DOSSWAP.EXE)

raţii au fost cuprinse în aceeaşi noţiune de *formatare*.

De la apariţia primului harddisk situaţia s-a schimbat. Împărţirea harddisk-ului în sectoare este realizată, de regulă, de producător, încă înainte de livrare. În sarcina utilizatorului mai rămîne aşadar doar crearea structurii logice pentru MS-DOS pentru a putea utiliza harddisk-ul. Această operaţie este denumită *formatare*, nu în utlimul rînd, deoarece comanda cu care este realizată este denumită, ca şi înainte *format*. Dar cum s-ar denumi atunci prima operaţie de pregătire? Producătorii îi spun - cum este şi de aşteptat - *formatare*.

Pentru a mai atenua confuzia împărţirea în sectoare mai este denumită adesea şi *formatare low-level*, sau *formatare fizică*. Deoarece comanda MS-DOS *format* nu poate realiza această operaţie, atunci cînd unitatea desemnată este un harddisk, pentru a o realiza este necesar un program special sau o procedură BIOS. În orice caz nici una nici cealaltă metodă nu sînt normate. Din fericire această operaţie trebuie executată foarte rar de utilizatori, majoritatea dintre ei nefiind puşi practic niciodată în situaţia de a-şi formata fizic harddisk-ul.

Dacă împărţirea în sectoare nu este realizată cu comanda *format*, este de presupus că această comandă realizează "celălalt fel" de formatare, creînd structura logică necesară pentru sistemul de operare MS-DOS. Tocmai acesta este cazul. Structura prezentată în fig. 1 constituie baza gestiunii fişierelor pe un disc. Aproape auxiliară este posibilitatea ca sistemul de operare MS-DOS să se autoîncarce de pe disc.

Un sector important

Cheia întregului disc se află în primul sector. Acest sector este denumit *sectorul de boot*, denumire care vine de la termenul englezesc *to boot* care desemnează pornirea sistemului de operare. (De aici şi barbarismele boot-area, boot-abil ş.a.m.d.). Dar chiar şi în cazul în care discul nu conţine nici un sistem de operare, şi deci calculatorul nu se poate "boot-a" de pe el, acest sector este foarte important.

Motivul pentru care acest sector este atît de important poate fi observat din tabela 1. Toate datele care descriu formatul discului, şi cu ajutorul cărora sistemul de operare MS-DOS poate utiliza discul, sînt stocate în *sectorul de boot*. Datele se ascund în spatele prescurtării *DPB* (Disk Parameter Block). Este vorba despre descrierea structurii logice a discului. Numai cu ajutorul acestor date sistemul de operare poate găsi *primul director* de pe disc, şi numai prin accedarea primului director este în stare MS-DOS să găsească fişierele. După cum se poate observa, sectorul de boot este cheia întregului disc.

În sectorul de boot se găsesc însă pe lîngă DPB, şi alte date importante. Importantă este şi mica procedură de încărcare, ce serveşte la pornirea sistemului de operare. Pe harddisk-uri mai există în sectorul de boot şi o altă structură de date, *tabela de partiţionare*. Dacă aţi utilizat un harddisk cu capacitatea mai mare de 32 MByte, cu un sistem de operare anterior lui MS-DOS versiunea 4.0, cuvîntul *partiţie* desemnează cu siguranţă o noţiune cunoscută.

Cu versiunile mai vechi de MS-DOS, un harddisk nu putea avea o capacitate de memorare mai mare de 32 MByte. Dar întru-cît multe harddisk-uri sînt mai mari, MS-DOS oferă o cale ocolitoare pentru a putea utliza întreaga capacitate a discului. Harddisk-ul este împărţit

Adresa	Lungime	Conţinut
0	3	Comandă de salt la începutul procedurii de boot
3	8	Numele producătorului şi numărul de versiune
11	2	Numărul de bytes dintr-un sector
13	1	Numărul de sectoare dintr-un cluster
14	2	Numărul de sectoare rezervate
16	1	Numărul de FAT-uri
17	2	Numărul de intrări în primul director
19	2	Numărul de sectoare de pe disc
21	1	Tipul discului
22	2	Numărul de sectoare din FAT
24	2	Numărul de sectoare per pistă
26	2	Numărul capetelor de scriere/citire
28	2	Numărul de sectoare ascunse
30	416	Procedura de boot
446	16	Tabela de partiţionare (numai la harddisk-uri)
462	50	Extensie la procedura de boot

Tabela 1: Structura sectorului de boot

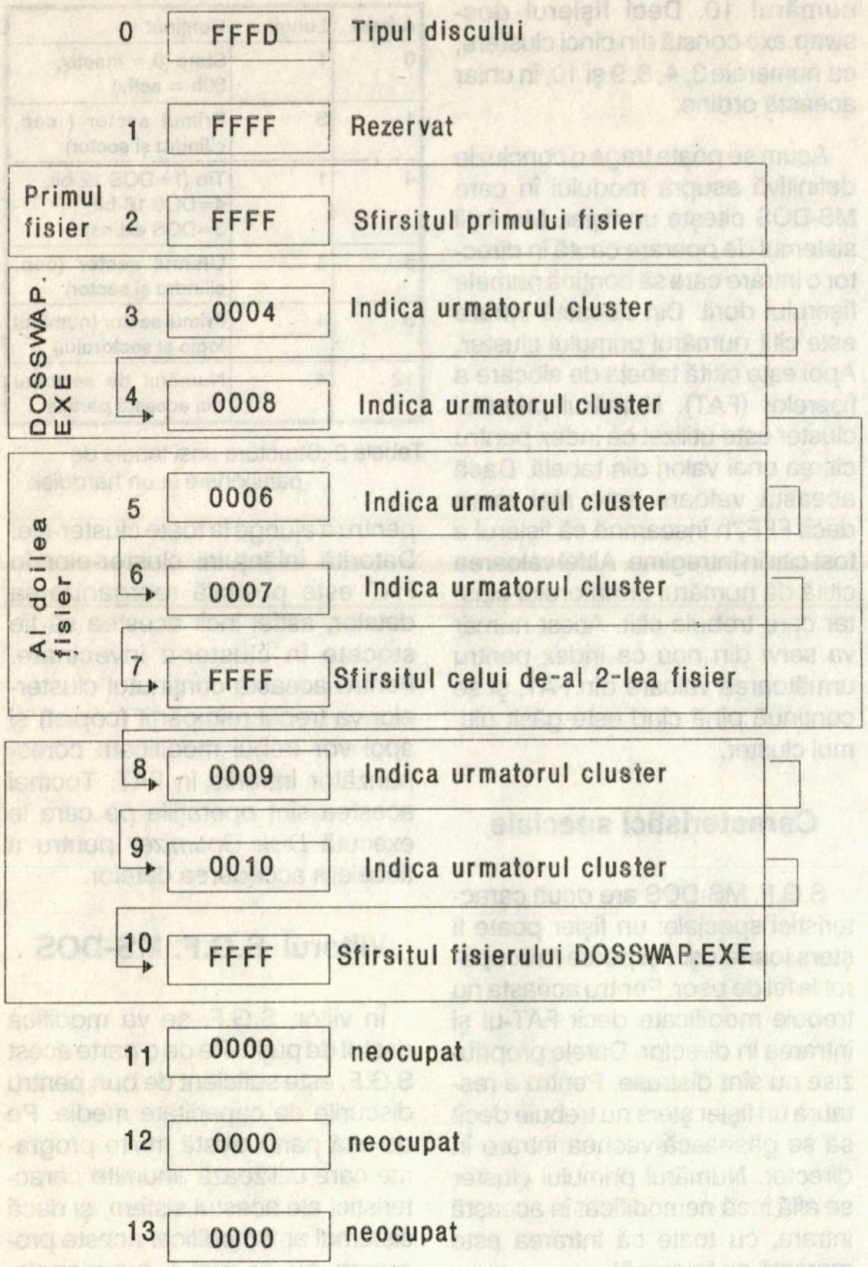


Fig. 3: Exemplu de structură a unui FAT (File Allocation Table)

în mai multe unităţi logice, fiecare unitate putînd avea capacitatea maximă de 32 MByte. Împărţirea unui harddisk în unităţi logice este denumită *partiţionare*. În tabela 2 este prezentată structura de date, din sectorul de boot, care stabileşte dimensiunea şi poziţia partiţiilor.

S-a putut observa cum este împărţit un harddisk şi cum poate găsi sistemul de operare primul director. Acest prim director este denumit adesea şi *root* (rădăcină) deoarece constituie baza structurii arborescente a directoarelor de pe disc. Legăturile dintre directoare şi dintre directoare şi fişere sînt conţinute în cuprinsul directoarelor. Fig. 2 arată cum este construită o astfel de intrare în director.

La începutul intrării se află numele directorului, respectiv al fişierului. Acest nume este constituit din 11 caractere, fără punct între nume şi extensie. După nume urmează aşa-numitul *attribute byte*, care conţine o informaţie importantă: cu ajutorul unui bit din acest byte se stabileşte dacă intrarea reprezintă un director sau un fişier.

Tot în această intrare sînt stocate date referitoare la dimensiunea fişierului şi la data şi ora creării. Aceste date nu sînt necesare S.G.F.-ului, el le memorează dar nu le utilizează. Dimpotrivă, cîmpul *Primul Cluster* este deosebit de important pentru S.G.F. MS-DOS.

Un *cluster* este o unitate logică care constă dintr-unul sau mai multe sectoare fizice învecinate. Aceasta înseamnă că sectoarele sînt grupate în mici entităţi denumite clustere. În mod obişnuit un cluster conţine un sector pe o dischetă şi patru sectoare pe un harddisk.

Există două motive care justifică această grupare: pe de o parte timpul este mai eficient utilizat dacă S.G.F. scrie mai multe sectoare odată. Acest mod de lucru conduce însă la o risipă de spaţiu de memorare. Astfel, de exemplu, un fişier cu dimensiunea de doar 10 Byte ocupă un cluster, şi deci 4 sectoare.

Al doilea motiv care justifică gruparea cîntăreşte mai greu: MS-DOS nu poate trata mai mult de cca. 65.000 de unităţi (sectoare, respectiv clustere) pe un harddisk. Cu aceasta - la o dimensiune tipică de sector de 512 Byte dimensiunea maximă a unui harddisk este limitată la aproximativ 32 MByte. Dacă sectoarele sînt grupate în unităţi mai mari, atunci se extinde şi această valoare limită. Cu patru sectoare per cluster dimensiunea maximă a unui harddisk devine 128 MByte. Această limită era neînchipuit de mare în urmă cu cîţiva ani, astăzi însă ea poate fi atinsă cu uşurinţă. Sîntem curioşi dacă noile versiuni MS-DOS vor ridica această limită.

Dar să ne întoarcem la cîmpul *Primul Cluster*. Valoarea acestui cîmp stabileşte poziţia de început pe harddisk a fişierului. Cu toate că de regulă un fişier ocupă mai multe cluster-e, în director nu există memorat decît primul cluster. Cum se regăsesc celelalte cluster-e ale fişierului?

Răspunsul este dat de structura de date concepută special pentru acest scop. Ea este denumită *File Allocation Table* (tabela de alocare a fişierelor), mai scurt *FAT*. Această structură este atît de importantă încît pe fiecare disc ea este stocată în dublu exemplar.

Sistemul de operare MS-DOS se îngrijeşte ca ambele copii să se găsească în aceeaşi stare, fapt care poate fi verificat cu comanda *Chkdsk*. În sarcina acestui program cade verificarea conţinutului celor două FAT-uri, atît una faţă de cealaltă cît şi în sine. Mesajele de eroare date de Chkdsk relevă date inconsistente, sau eronate, în una sau în ambele copii FAT.

În fig. 3 este dat un mic exemplu al unui FAT. Aşa cum se poate observa, pentru fiecare cluster există cîte o intrare în FAT. Numărul cluster-ului serveşte ca index în FAT. În funcţie de faptul dacă, şi cum, un cluster este ocupat, intrarea ia diferite valori.

Valoarea 0 desemenează un cluster liber, care poate fi utilizat la crearea unui nou fişier, respectiv a unui nou director. Valorile de la FFF0h pînă la FFF6h sînt rezervate, iar valoarea FFF7h desemnează un cluster defect, în care nu pot fi memorate date. Valorile de la FFF8h pînă la FFFFh specifică faptul că este vorba despre ultimul cluster al unui fişier. Toate celelalte valori au o semnificaţie simplă: conţin numărul următorului cluster al fişierului.

Dacă urmăriţi intrarea în director din fig. 2, veţi observa că în cîmpul Primul Cluster este memorată valoarea 0003h. În a treia intrare în FAT (fig.3) veţi găsi valoarea 0004h. Aceasta înseamnă că al doilea cluster al fişierului dosswap.exe este cluster-ul numărul 4. În a patra intrare în FAT veţi găsi valoarea 0008h, deci cluster-ul numărul 8 este al treilea cluster al fişierului. FAT-ul este parcurs în continuare în acest mod pînă cînd este întîlnită valoarea FFFFh. Aceasta înseamnă că ultimul cluster al fişierului este cluster-ul numărul 10. Deci fişierul dosswap.exe constă din cinci clustere, cu numerele 3, 4, 8, 9 şi 10, în chiar această ordine.

Acum se poate trage o concluzie definitivă asupra modului în care MS-DOS citeşte un fişier. Mai întîi sistemul de operare caută în director o intrare care să conţină numele fişerului dorit. Din această intrare este citit numărul primului cluster. Apoi este citită tabela de alocare a fişerelor (FAT). Numărul primului cluster este utilizat ca index pentru citirea unei valori din tabelă. Dacă această valoare este mai mare decît FFF7h înseamnă că fişierul a fost citit în întregime. Altfel valoarea citită dă numărul următorului cluster care trebuie citit. Acest număr va servi din nou ca index pentru următoarea valoare din FAT, şi se continuă pînă cînd este găsit ultimul cluster.

Caracteristici speciale

S.G.F. MS-DOS are două caracteristici speciale: un fişier poate fi şters foarte uşor şi poate fi recuperat la fel de uşor. Pentru aceasta nu trebuie modificate decît FAT-ul şi intrarea în director. Datele propriuzise nu sînt distruse. Pentru a restaura un fişier şters nu trebuie decît să se găsească vechea intrare în director. Numărul primului cluster se află încă nemodificat în această intrare, cu toate că intrarea este marcată ca "ştearsă".

Pornind de la acest prim cluster, cu ajutorul FAT-ului pot fi regăsite toate celelalte cluster-e. Cu singura condiţie ca după ştergere niciunul dintre aceste cluster-e să nu fi fost alocat altui fişier. Aceasta este tehnica utilizată de utilitarele care recuperează fişierele şterse.

A doua caracteristică a sistemului de fişiere este mai puţin avantajoasă. Deoarece datele nu trebuie stocate neapărat grupat, în anumite condiţii, ele sînt "împrăştiate" în cluster-ele de pe disc. Citirea unui astfel de fişier necesită relativ mult timp, deoarece capetele de citire trebuie să se mişte des şi departe,

Adresă	Lungime	Conţinut
0	1	Stare (0 = inactiv, 80h = activ)
1	3	Primul sector (cap, cilindru şi sector)
4	1	Tip (1=DOS 12-bit, 4=DOS 16-bit, 5=DOS extins)
5	3	Ultimul sector (cap, cilindru şi sector)
8	4	Primul sector (numărul logic al sectorului)
12	4	Numărul de sectoare din această partiţie

Tabela 2 :Structura unei tabele de partiţionare la un harddisk

pentru a ajunge la toate cluster-ele. Datorită înlănţuirii cluster-elor în FAT este posibilă reorganizarea datelor, astfel încît acestea să fie stocate în cluster-e învecinate. Pentru aceasta conţinutul cluster-elor va trebui reîmpărţit (copiat) şi apoi vor trebui modificate corespunzător intrările în FAT. Tocmai acestea sînt operaţiile pe care le execută *Disk Optimizer* pentru a accelera accedarea datelor.

Viitorul S.G.F. MS-DOS

În viitor, S.G.F. se va modifica destul de puţin. Pe de o parte acest S.G.F. este suficient de bun pentru discurile de capacitate medie. Pe de altă parte există multe programe care utilizează anumite caracteristici ale acestui sistem, şi dacă sistemul ar fi modificat aceste programe nu ar mai fi funcţionale. Aceasta nu înseamnă însă că nu este de aşteptat nici o noutate.

Astfel există, de exemplu, o componentă MS-DOS mai puţin cunoscută, care se numeşte *Network Redirector* şi care permite cuplarea unui S.G.F. dorit în MS-DOS. Există deja cîteva domenii în care această facilitate este utilizată: unităţile CD, unele reţele şi diferite alte periferice pot fi adresate prin această interfaţă. În anii următori vor fi cu siguranţă noutăţi în acest domeniu.

(R.M.)

La nivelul cel mai de jos

Turbo-Prolog - Partea a treia

În cea de a treia şi ultima parte a cursului ne vom ocupa de modul în care se poate programa apropiat de codul maşină şi de cuplarea unor proceduri externe într-un program Prolog.

Şi în Turbo-Prolog sînt posibile programarea Low-Level (la nivelul de jos, apropiat maşină) şi accesul la nivelul sistemului de operare. Ca şi în celelalte limbaje de programare Turbo, există o serie de predicate (funcţii) care permit accesul la nivel DOS şi la nivelul hard. Pentru apelul unor comenzi MS-DOS poate fi utilizat predicatul "system". Pentru manipularea datei şi orei sistem pot fi utilizate predicatele "date" şi "time". În plus, începînd din versiunea 2.0, cu ajutorul predicatelor "envsymbol" şi "comline" pot fi preluate variabilele de mediu şi parametrii liniei de comandă. Predicatul "system" oferă următoarele două posibilităţi: "system (Comandă DOS)" şi "system (Comanda DOS, Reset Video, Errorlevel)".

În cazul primei variante comanda DOS este transmisă predicatului ca şir, de exemplu: "system ("del *.*")". Dacă acest şir este vid, atunci se încarcă o copie a fişierului COMMAND.COM din care se poate reveni în Prolog dacă se tastează "exit".

În cazul variantei extinse pe lîngă comanda DOS se poate stabili prin intermediul argumentului "Reset Video" dacă se doreşte, sau nu, refacerea imaginii curente ecran la terminarea execuţiei comenzii DOS. "Reset Video = 1" va determina refacerea ecranului, lucru care nu se va întîmpla dacă "ResetVideo = 0". În plus codul de retur al comenzii apelate este încărcat în variabila "Errorlevel" şi poate fi prelucrat mai departe.

Cu ajutorul predicatelor "time" şi "date" pot fi manipulate cu uşurinţă timpul şi data sistemului. Sintaxa de apel este: "time (Ora, Minut, Secunda, Sutime)" sau "date (An, Luna, Zi)". Variabilele utilizate trebuie fie să fie toate legate, fie să fie toate libere. Dacă variabilele sînt libere, atunci în ele vor fi încărcate valorile curente din sistem; dacă variabilele sînt legate atunci se va modifica timpul sau data sistemului.

Predicatul "envsymbol" permite preluarea valorii unei variabile din mediul actual al programului. Sintaxa predicatului "envsymbol" este următoarea: "envsymbol (EnvVar, Valoare)". "EnvVar" stabileşte numele variabilei de mediu a cărei valoare va fi returnată ca şir în variabila "Valoare". În cazul în care variabila specificată nu există, predicatul "envsymbol" eşuează.

Pentru a putea accede la parametrii liniei de comandă poate fi utilizat predicatul "comline", care poate fi apelat cu sintaxa "comline (Parametri)."

În urma apelului parametrii liniei de comandă sînt legaţi de variabila şir "Parametri". Acest predicat a fost compilat şi transformat în program executabil (.EXE), deoarece în Turbo-Prolog parametrii nu pot fi transmişi la fel ca şi în Turbo-Pascal, de exemplu.

Programul de exemplu din fig. 1, arată modul în care pot fi utilizate predicatele prezentate.

Programul citeşte linia de comandă şi interpretează parametrii acesteia ca pe o comandă DOS care trebuie executată. Dacă în linia de comandă nu se specifică nici un parametru atunci se încarcă o copie a lui "COMMAND.COM". Programul cronometrează durata de timp pentru care copia a fost încărcată şi la revenire afişează acest interval de timp pe monitor.

Suplimentar se afişează valoarea variabilei de mediu PATH. Pentru ca programul să poată prelua parametrii, el trebuie compilat anterior şi transformat în fişier executabil (.EXE).

```
predicates
   gettime (real)
   write_command (string)
   showpath
   do_DOS_Command

clauses
   gettime(Sec):-
      Time (H,M,S,S100),
      Sec = 3.6E2*H + 60*M + S + S100/100.

   showpath:-
      envsymbol("PATH",Path),
      write("\n\n Variabila de mediu PATH"),
      write("\n este pozitionata pe", Path,"\n\n").

   write_command(""):-
      write_command("COMMAND.COM").
   write_command(X):-
      write (X," a fost executata!\n\n").

   do_DOS_Command:-
      gettime(SecStart),
      comline(ComandaDOS),
      system (ComandaDOS,1,_),
      write_command(ComandaDOS),
      gettime (SecEnd),
      Secunde = SecEnd-SecStart,
      writef("DOS a fost activ %-1.0f secunde!",Secunde),
      showpath,
      write("Terminati apasind o tasta ..."),
      readchar(_).

goal
   do_DOS_Command.
```

Fig. 1

Turbo-Prolog permite programatorului şi accelerarea directă a întreruperilor calculatorului. Cu ajutorul predicatului "bios" pot fi executate toate întreruperile calculatorului. Acest predicat poate fi folosit cu una din sintaxele următoare: "bios (Interruptnummer, Input Register, OutputRegister)" sau "bios (Interruptnummer, InputRegistrer, OutputRegister, Flags)". "Interruptnummer" desemnează numărul întreruperii de executat. "InputRegister" şi "OutputRegister" sînt date structurate de tipul "regdom". Acest tip de date va fi definit în felul următor:

```
domains
regdom = reg (integer, integer, integer,
   integer, integer, integer, integer, integer)
```

Astfel nu mai este necesară o redefinire a lui în "domains".

Cu "reg (...)" sînt desemnaţi regiştrii individuali ai procesorului principal din care se citeşte sau în care se scrie.

```
/* Registri de intrare */
reg (AXi, BXi, CXi, DXi, Sli, Dli, DSi, ESi)
/* Registri de iesire */
reg (AXo, BXo, CXo, DXo, Slo, Dlo,
   DSo, ESo)
```

Un apel al lui "bios" ar putea arăta în modul următor:

```
bios (Interruptnummer,
reg (AXi, BXi, CXi, DXi, Sli, Dli,
   DSi, ESi),
reg (AXo, BXo, CXo, DXo, Slo, Dlo,
   DSo, ESo)).
```

"AXi", "BXi", "..." desemnează valorile de intrare pentru execuţia întreruperii. Valorile registrelor "AXo", "BXo", "..." sînt returnate de predicat după apelul întreruperii. Trebuie avută grijă ca toate variabilele "?Xi" să fie legate şi respectiv toate variabilele "?Xo" să fie libere.

Opţional predicatul "bios" mai poate conţine şi un al patrulea parametru "flag" prin intermediul căruia pot fi returnate flag-urile procesorului. Figura 2 arată modul în care sînt împachetate flag-urile procesorului într-o variabilă întreagă. Prescurtările utilizate au următoarea semnificaţie:

U = flag de depăşire
R = flag de direcţie
I = flag de întrerupere
E = flag de pas cu pas
V = flag de semn
N = flag de nul
H = carry auxiliar
P = paritate
C = carry

Care flag-uri sînt poziţionate şi după care întrerupere, se poate citi dintr-un manual DOS.

Aşa cum a fost descris, în Turbo-Prolog există posibilitatea apelării unei întreruperi. În plus programatorul poate utiliza următoarele predicate pentru programarea low-level:

```
memword (Segment, Offset, Word)
membyte (Segment, Offset, Byte)
portbyte (PortNummer, Byte)
ptr-dword (StringVar, Segment, Offset)
```

"membyte" şi "memword" permit accesul la octeţi individuali. Dacă al treilea argument al predicatului este liber încă în momentul apelului, atunci celula de memorie specificată prin "Segment: Offset" este citită şi valoarea ei este legată de această variabilă, altfel conţinutul variabilei este scris în respectiva celulă de memorie.

Predicatul "portbyte" scrie, respectiv citeşte, un byte la portul specificat.

Cu ajutorul predicatului "ptr-dword" poate fi stabilită adresa absolută a unei variabile şir. Trebuie avută grijă în acest caz deoarece terminatorul de şir nu este caracterul null, la fel ca în Turbo-Pascal. Dacă variabila "StringVar" este liberă atunci de ea este legată valoarea "Segment: Offset", altfel Segment şi Offset sînt legate de valorile corespunzătoare. Adresa absolută a celulei de memorie se poate obţine apoi cu relaţia: "Segment * 16 + Offset".

■	■	■	■	U	R	I	E	V	N	■	H	■	P	■	C
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Fig. 2

Programul din fig. 3 demonstrează modul în care pot fi utilizate predicatele low-level. Programul de exemplu comută mai întîi, printr-un acces direct la memorie, tasta "PrtScr". Apoi calculatorul poate fi accelerat sau frînat între anumite limite prin modificarea ratei de refresh. În cele din urmă cu ajutorul funcţiei "43h" a întreruperilor DOS este modificat "Hidden-Flag"-ul unui fişier. Dacă fişierul specificat este vizibil în director, atunci este ascuns şi invers.

Turbo-Prolog pune la dispoziţia programatorilor un mare număr de funcţii high-level. Uneori este necesar totuşi să se insereze într-un program şi segmente realizate cu alte limbaje de programare procedurale. Pentru acest scop compilatorul dispune de interfeţe pentru Assembler, C, Fortran şi Pascal. În continuare va fi clarificat modul de utilizare al unor proceduri externe

```
predicates
   ptrscroff
   changerefresh
   cangehideflag

goal
   ptrscroff,
   changerefresh,
   changehideflag.

clauses

ptrscroff:-
   membyte(0,1280,1),
   write("PtrScr - Tasta dezactivata!\n").

changerefresh:-
   write("Noua rata de refresh ( > =3): "),
   readint(X),
   port_byte(67,84),
   port_byte(65,X).

changehideflag:-
   write("Modificare hiddenflag !\n"),
   write("Nume fisier: "),
   readln(S),
   ptr_dword(S,DSi,DXi),
   bios(33,reg($4300,0,0,DXi,0,0,DSi,0),
      reg(_,_,CX,_,_,_,_,_)),
   bitxor(CX,2,CXi),
   bios(33,reg($4301,0,CXi,DXi,0,0,DSi,0),
      reg(_,_,_,_,_,_,_,_)).
```

Fig. 3

```
IDIV_PROC SEGMENT BYTE
        PUBLIC 'CODE'
   ASSUME CS:IDIV_PROC
   PUBLIC  idiv_0

idiv_0  proc far
        push bp
        mov bp,sp
        xor dx,dx
        mov ax,[bp]+16
        mov cx,[bp]+14
        idiv cx
        lds si,dword ptr [bp]+10
        mov [si],ax
        lds si,dword ptr  [bp]+6
        mov [si],dx
        pop bp
        ret 12
idiv_0  endp

IDIV_PROC ENDS
END
```

Fig. 4

prin intermediul a trei mici exemple în Assembler şi C.

Toate procedurile care sînt formulate într-un alt limbaj, sînt apelate în Turbo-Prolog ca şi predicate globale. La declararea unui modul extern trebuie specificat explicit limbajul folosit prin intermediul cuvîntului "language". Definiţia unui predicat global va arăta în modul următor:

```
global predicates
test (integer, real)-
   (i,o), (o,i) language asm
```

Variantele de apel admisibile trebuie specificate înaintea cuvîntului cheie "language", "i" este utilizat pentru Input şi "o" pentru "Output". Un predicat poate avea în Turbo-Prolog mai multe variante de tip şi de apel. Din această cauză diferitele proceduri pentru acelaşi predicat trebuie să conţină un nume, urmat de o linie de subliniere ('-') şi apoi să fie numerotate de la 0 în sus; de exemplu pentru predicatul "test" vor exista "test-0" şi "test-1". Pentru adresarea memoriei Turbo-Prolog utilizează pointeri de 32 de biţi şi din această cauză şi în procedurile externe apelurile şi retururile din subprograme trebuie să fie de tipul "FAR".

Următorul exemplu prezintă o rutină în Assembler care efectuează o împărţire întreagă cu valorile parametrilor furnizaţi şi

```
global predicates
   idiv(integer,integer,integer,integer)
   - (i,i,o,o) language asm
goal
   write("Deimpartit: "),
   readint(Deimpartit),
   write("Impartitor: "),
   readint(Impartitor),
   idiv(Deimpartit,Impartitor,Rezultat,Rest),
   write("Rezultat: ",Rezultat," Rest: ",Rest),
   nl.
```

Fig. 5

livrează ca parametri de ieşire cîtul şi restul împărţirii. În fig. 4 este prezentat programul corespunzător scris în Assembler.

Modul de construire a stivei pentru rutina "idiv" este prezentat în fig. 5. În programul Prolog din fig. 6 este arătat modul de utilizare al modulului extern. Pentru a cupla cele două bucăţi de program sursele "IDIV.ASM" şi "ASM-PRO.PRO" vor trebui mai întîi compilate pentru a se obţine fişierele obiect (.OBJ).

După compilare sînt disponibile fişierele "IDIV.OBJ", "ASM_PRO.OBJ" şi "ASM_PRO.SYM". Cu comanda: "TLWK INIT ASM_PRO IDIV ASM_PRO.SYM, ASMMIXED,, PROLOG" se realizează editarea de legături şi se obţine programul executabil "ASMMIXED.EXE".

Turbo-Prolog poate fi completat şi cu Turbo-C un mediu de programare extrem de puternic. Prin utilizarea de rutine Turbo-C facilităţile limbajului Prolog pot fi mult extinse. Două exemple (fig. 7 şi 8) lămuresc modul de conlucrare al celor două limbaje de programare.

În primul exemplu se arată, cu ajutorul unui modul C simplu, cum poate fi implementat un predicat cu mai multe forme de apel. Pentru aceasta pentru fiecare din cele trei forme de apel "i,i,o", "i,o,i" şi "o,i,i" este implementată cîte o rutină C. Programul apelant Turbo-Prolog utilizează apoi predicatul global cu diferitele forme de apel.

La compilarea modulelor trebuie avute în vedere următoarele aspecte:

```
sub_0(int A, int B, int *C) /* (i,i,o) */
{
   *C = A - B;
}

sub_1(int A, int *B, int C) /* (i,o,i) */
{
   *B = A - C;
}

sub_2(int *A, int B, int C)  /* (o,i,i) */
{
   *A = B + C;
}
```

Fig. 6

1) Numele funcţiilor C, care vor fi apelate de Turbo-Prolog trebuie să se termine cu o liniuţă de subliniere şi un număr de secvenţă.
2) Modulele corespunzătoare Turbo-C trebuie compilate cu modelul de memorie "large".
3) Dacă se utilizează Turbo-Prolog Version 1.1 un modul C cu variabile registru nu poate fi compilat, aceasta înseamnă că trebuie poziţionat comutatorul "use registers variables", al compilatorului, pe OFF.
4) Comutatorul compilatorului "Generate underbars" trebuie poziţionat pe OFF.

La editarea de legături trebuie avute în vedere următoarele aspecte:

1) Modulul de iniţializare Turbo-Prolog "INIT.OBJ" trebuie specificat ca prim parametru.
2) Dacă se utilizează funcţii de bibliotecă Turbo-C, la editarea de legături trebuie dată şi biblioteca "CL.LIB". Pentru operaţii cu numere reale sînt necesare şi bibliotecile EMU.LIB şi MATHL.LIB.

Apelul editorului de legături va arăta în modul următor:

```
TLWK INIT <modul principal TP>
   <fis.obiect>
   <modul principal TP.SYM>,
   [nume fis. EXE],,
   [biblioteci] PROLOG +
   [EMU+MATHL]+ CL
```

(continuare în pag. 51)

C++, partea a II-a

Diversitate virtuală

Pe lîngă moştenire şi încapsulare, polimorfismul este cea mai importantă caracteristică de bază a obiectelor. Acest articol arată cum se utilizează.

Înainte de a analiza mai amănunţit avantajele polimorfismului urmează un scurt paragraf relativ la funcţii virtuale.

Prin introducerea funcţiilor virtuale, listingul EX5.CPP din articolul precedent se simplifică. În exemplu sînt declarate metodele pentru două clase diferite. Prima clasă "Semn" nu memorează informaţii despre fond. Clasa derivată "Semn2" memorează conţinutul şi atributul ecran din poziţia X,Y, înainte de a afişa acolo un nou caracter.

Deoarece clasa derivată moşteneşte metodele clasei de bază, sînt imediat disponibile toate metodele pentru clasa derivată "Semn2". Dar, pentru acest obiect modificat, nu toate metodele lucrează. Acest lucru este evidenţiat mai cu seamă de metoda "Ştergere", care funcţionează diferit de la o clasă la alta. Clasa "Semn" nu ţine seama de conţinutul precedent al ecranului. Metoda Semn2::Ştergere( ) în schimb reafişează vechiul conţinut al ecranului în poziţia x,y. Deoarece şi alte metode (ca de exemplu: "Mişcare( )") se bazează pe metoda "Ştergere" corespunzătoare, în Listingul EX5 nu a fost posibilă preluarea nerestricţionată a metodelor din clasa "Semn". În cadrul clasei "Semn 2" a fost nevoie să se recurgă fie la metoda din "Semn::Citire( )", "Semn::Ştergere( )" şi "Semn::Mişcare( )" au fost definite ca metode virtuale. Acest lucru înseamnă că în cadrul clasei "Semn2" aceste metode sînt înlocuite cu metode noi. Ele nu folosesc metodele clasei de bază. Prin aceasta, listingul devine mult mai clar şi mai scurt. Obiectele şi metodele pot fi comparate cu propoziţii ce conţin subiect şi predicat. Obiectele (clasele) corespund subiectului iar metodele predicatului. Să considerăm propoziţiile:

"Apa curge", "Nasul curge", "Timpul curge". Se constată cu uşurinţă că metoda "curge( )" este aplicabilă unor obiecte diferite. Abia în legătură cu o clasă apare un mesaj inteligibil. Doar este o mare diferenţă între faptul că apa curge sau nasul curge. În schimb mesajul "Ora curge" se exprimă asemănător cu obiectul timp. Această prezentare plastică ajută mult la înţelegerea polimorfiei.

```
1   // Listing exemplu nr. 6
2   // functii virtuale
3
4   #include <conio.h>
5   #include <stdio.h>
6
7   class Semn      // definitia clasei Semn
8   {
9     int X, Y;    // coordonate
10    public:
11    char valoare; // valoare alfanumerica
12    char attr;    // atribut ecran
13
14    Semn (int InitX, int InitY)
15    {
16      X = InitX;
17      Y = InitY;
18    }
19
20    // Functii-element publice
21
22    int Semn::GetX ( ) { return X; }
23    int Semn::GetY ( ) { return Y; }
24    void Semn::Position (int PosX, int PosY)
25    {
26      X = PosX; Y = PosY;
27    }
28    void Semn::Citire ( )
29    {
30     gettext (X, Y, X, Y, &valoare);
31    }
32    void Semn::Scriere (char Sign)
33    {
34     Citire ( );
35     valoare = Sign;
36     puttext (X, Y, X, Y, &valoare);
37    }
38    void Semn::Modificare (char Sign)
39    {
40     valoare = Sign;
41     puttext (X, Y, X, Y, &valoare);
42    }
43    void Semn::Indicare ( )
44    {
45     puttext (X, Y, X, Y, &valoare);
46    }
47    virtual void Semn::Stergere ( )
48    {
49     valoare = ' ';
50     Indicare ( );
51    }
52    virtual void Semn::Miscare (int NewX, int NewY)
53    {
54     char memo_valoare;
55     memo_valoare = valoare;
56     Stergere ( );
57     valoare = memo_valoare;
58     Position (NewX, NewY);
59     Indicare ( );
60    }
61    void Semn::Copiere (int NewX, int NewY)
62    {
63     Position (NewX, NewY);
64     Indicare ( );
65    }
66    void Semn::Seteaza_attr (char attr_nou)
67    {
68     attr = attr_nou;
69     Indicare ( );
70    }
71  };
72
73  // definitia clasei Semn2
74
75  class Semn2 : public Semn
76  {
77   protected:
78   char valoare_veche;
79   char attr_vechi;
80
81   public:
82   Semn2 (int InitX, int InitY) : Semn (InitX, InitY)
83   {
84    valoare_veche = 0x0;
85    attr_vechi = 0x0;
86   }
87
88   // Functii-element publice
89
90   void Semn2::Citire ( )
91   {
92    gettext (GetX( ), GetY( ), \
93          GetX( ), GetY( ), &valoare_veche);
94   }
95   void Semn2::Stergere( )
96   {
97    valoare = valoare_veche;
98    attr = attr_vechi;
99    Indicare ( );
100  }
101  void Semn2::Miscare (int NewX, int NewY)
102  {
103   char memo_valoare;
104   char memo_attr;
105
106   memo_valoare = valoare; valoare = valoare_veche;
107   memo_attr = attr; attr = attr_vechi;
108   Indicare ( );
109   valoare = memo_valoare;
110   attr = memo_attr;
111   Position (NewX, NewY);
112   Citire ( );
113   Indicare ( );
114  }
115 };
116
117 int main ( )
118 {
119  int X1, Y1, s, z;
120
121  X1 = 1; Y1 = 1;
122  Semn BSemn (X1, Y1);
123  BSemn.Scriere ('+');
124  BSemn.Seteaza_attr (BLUE);
125  for (s = 0; s < 25; s++)
126   for (z = 0; z < 80; z++)
127    BSemn.Copiere (X1+z, Y1+s);
128  X1 = 4; Y1 = 10;
129  Semn2 ASemn (X1, Y1);
130  ASemn.Scriere ('A');
131  ASemn.Seteaza_attr (WHITE);
132  printf ("\n Apasati o tasta ... ");
133  getch ( );
134  ASemn.Miscare (X1 + 4, Y1 - 4);
135  getch ( );
136  ASemn.Modificare ('B');
137  getch ( );
138  ASemn.Stergere ( );
139  getch ( ); clrscr ( );
140  return 0;
141 }
```

EX6.CPP

Transparenţă prin diversitate de forme

Asupra unor obiecte diverse pot fi aplicate aceleaşi metode. Sigur că nu este o mare diferenţă între deplasarea pe ecran a unui cerc sau a unui dreptunghi. Metoda "Deplasare( )" poate fi aplicată unor metode diverse.

Din punct de vedere al conţinutului, listingurile EX71.CPP pînă la EX73.CPP sînt legate. Primul listing este gîndit ca o introducere în tema polimorfism. În listingul EX71.CPP sînt prezentate cele mai importante metode pentru obiectele "Semn", "Linie", "Pătrat" şi "Dreptunghi", şi anume "Ştergere( )", "Indicare( )", "Mişcare( )". Clasa "Linie" este o clasă derivată a clasei "Semn". Această clasă este o linie orizontală, ce se întinde pe direcţia x. Clasa "Linie2" este o linie verticală. Aceasta moşteneşte toate metodele clasei "Linie". Numai metoda "Indicare( )" se deosebeşte de metoda "Linie::Indicare( )". Clasele "Pătrat" şi "Dreptunghi" sînt şi ele clase derivate. Deşi clasa "Pătrat" nu se prezintă ca un pătrat pe ecran, lungimea ei pe direcţia x coincide cu lungimea pe direcţia y. Pentru clasa "Pătrat" a trebuit definită numai metoda "Indicare". De remarcat că nu este necesar nici un atribut suplimentar. Pentru clasa "Dreptunghi" trebuiesc fixate lungimile pe direcţia x şi pe direcţia y.

În programul principal se exemplifică apelul metodelor pentru cele 3 clase numite. Pentru toate clasele se fac aceleaşi lucruri, în aceeaşi succesiune. Acesta este polimorfism.

Ştergerea obiectelor se preia şi pentru prim-plan. Atributul ecran are următoarea structură: biţii 0-3: culoarea prim-planului, biţii 4-6: culoarea fondului şi bitul 7: blinking (luminare intermitentă).

Cu metoda "Semn::BackColor( )" cei 3 biţi ai culorii de fond sînt deplasaţi. Această metodă este parcursă la orice ştergere, indiferent de clasa derivată la care tocmai se aplică. În cadrul metodei "Semn::Ştergere( )" se face trimiterea la metoda "Indicare( )", declarată virtuală. Funcţia Expand( ) furnizează un şir de caractere de lungime L, cu poziţii succesive pentru "valoare" şi "atribut".

Funcţia "gettext (x1,y1,x2,y2, buffer)" citeşte o fereastră ecran dintre colţul stînga sus (de coordonate x1,y1) şi colţul dreapta jos (x2,y2). În "buffer" se găseşte o porţiune a memoriei de reîmprospătare a ecranului şi anume (y2 - y1 + 1) linii şi (x2 - x1 + 1) coloane; lungimea totală trebuie înmulţită cu 2, deoarece pentru fiecare poziţie x,y pe ecran se stochează o valoare (1 octet) şi un atribut de culoare (1 octet).

```
1   // Exemplul 7.1
2   // obiecte pluriforme
3
4   #include <conio.h>
5   #include <stdio.h>
6   #include <stdlib.h>
7
8   void Expand (char w, char a, int l, char *t)
9   {
10   int i;
11
12   for (i=0; i<=2*l; i=i+2) {
13    t[i] = w; t[i+1] = a;
14   }
15  }
16
17  class Semn  // defineste clasa Semn
18  {
19   int x, y; // coordonatele
20
21   public:
22   char valoare; // valoare alfanumerica
23   char attr; // atribut ecran
24   Semn::Semn (int InitX, int InitY)
25   {
26    x = InitX;
27    y = InitY;
28   }
29   int Semn::GetX ( ) {return x;}
30   int Semn::GetY ( ) {return y;}
31   void Semn::Position (int PosX, int PosY)
32   {
33    x = PosX; y = PosY;
34   }
35   void Semn::Citire( )
36   {
37    gettext (x, y, x, y, &valoare);
38   }
39   virtual void Semn::Indicare ( )
40   {
41    puttext (x, y, x, y, &valoare);
42   }
43   void Semn::Scriere (char Sign)
44   {
45    valoare = Sign;
46    Indicare ( );
47   }
48   void Semn::Modificare (char Sign)
49   {
50    valoare = Sign;
51    Semn::Indicare ( );
52   }
53   void Semn::BackColor ( )
54   {
55    int Color;
56
57    if (attr > 128) {
58     Color = attr - 128;
59     Color = _rotr (Color, 4);
60     attr = Color + 16 * Color + BLINK;
61    }
62    else {
63     attr = _rotr (attr, 4);
64     attr = attr + 16 * attr;
65    }
66   }
67   void Semn::Stergere ( )
68   {
69    BackColor ( );
70    valoare = ' ';
71    Indicare ( );
72   }
73   void Semn::Miscare (int NewX, int NewY)
74   {
75    Stergere ( );
76    Position (NewX, NewY);
77    Indicare ( );
78   }
79   void Semn::Copiere (int NewX, int NewY)
80   {
81    Position (NewX, NewY);
82    Indicare ( );
83   }
84   void Semn::Set_attr (char fattr, char battr = 0,\
85                        char Blink = 0)
86   {
87    if (battr < 8)
88     attr = fattr + 16 * battr + Blink;
89    else
90     attr = fattr + Blink;
91   }
92  };
93
94  class Linie : public Semn
95  {
96   protected:
97   int lungime;
98   public:
99   Linie::Linie (int InitX, int InitY) \
100           : Semn (InitX, InitY)
101  {
102   lungime = 0;
103  }
104  int Linie::GetL ( ) { return lungime; }
105  void Linie::Indicare ( )
106  {
107   char temp[4096];
108   Expand (valoare, attr, lungime, temp);
109   puttext (GetX( ), GetY( ),\
110        GetX( )+lungime, GetY( ), &temp);
111  }
112  void Linie::Fix_lungime (int L)
113  {
114   lungime = L;
115  }
116 };
117
118 class Linie2 : public Linie
119 {
120  public:
121  Linie2::Linie2 (int InitX, int InitY)\
122          : Linie (InitX, InitY)
123  {
124   Linie::Fix_lungime (0);
125  }
126  void Linie2::Indicare ( )
127  {
128   char temp[4096];
129   Expand (valoare, attr, GetL( )*GetL( ), temp);
130   puttext (GetX( ), GetY( ),\
131        GetX( ), GetY( )+GetL( )-1, &temp);
132  }
133 };
134
135 class Patrat : public Linie
136 {
137  public:
138  Patrat::Patrat (int InitX, int InitY)\
139     : Linie (InitX, InitY)
140  {
141   Linie::Fix_lungime (0);
142  }
143  void Patrat::Indicare ( )
144  {
145   char temp [4096];
146
147   Expand (valoare, attr, GetL( )*GetL( ), temp);
148   puttext (GetX( ), GetY( ),\
149       GetX( )+GetL( )-1, GetY( )+GetL( )-1, &temp);
150  }
151 };
152
153 class Dreptunghi : public Semn
154 {
155  int LungX;
156  int LungY;
157
158  public:
159  Dreptunghi::Dreptunghi (int InitX, int InitY)\
160         : Semn (InitX, InitY)
```

```
{
 LungX = 0;
 LungY = 0;
}
void Dreptunghi::Indicare ( )
{
 char temp [4096];

 Expand (valoare, attr, LungX*LungY, temp);
 puttext (GetX( ), GetY( ),\
      GetX( )+LungX-1, GetY( )+LungY-1, &temp);
}
void Dreptunghi::Fix_lungime (int lx, int ly)
{
 LungX = lx;
 LungY = ly;
}
};

int main( )
{
 int x1, y1, s, z;
 // fond cu semn albastru
 x1 =1; y1 = 1;
 Semn BSemn (x1, y1);
 BSemn.Scriere (219);
 BSemn.Set_attr (BLUE);
 for (s=0; s<25; s++)
  for (z=0; z<80; z++)
   BSemn.Copiere (x1+z, y1+s);
 printf ("Apasati o tasta ...");
 getch ( );
 // Reprezinta si apoi sterge o linie
 x1 = 10; y1 = 10;
 Linie LSemn (x1, y1);
 LSemn.Set_attr (RED, BLUE);
 LSemn.Fix_lungime (60);
 LSemn.Scriere ('-');
 getch ( );
 LSemn.Stergere ( );
 getch ( );
 // Reprezinta si apoi sterge un patrat
 x1 = 10; y1 = 2;
 Patrat PSemn (x1, y1);
 PSemn.Set_attr (GREEN, BLUE);
 PSemn.Fix_lungime (20);
 PSemn.Scriere ('=');
 getch ( );
 PSemn.Stergere ( );
 getch ( );
 // Reprezinta si apoi sterge un dreptunghi
 x1 = 10; y1 = 2;
 Dreptunghi DSemn (x1, y1);
 DSemn.Set_attr (CYAN, BLUE);
 DSemn.Fix_lungime (40, 20);
 DSemn.Scriere (219);
 getch ( );
 DSemn.Stergere ( );
 getch ( );
 clrscr ( );
 return 0;
}
```

EX71.CPP

Desigur că e nevoie de un timp de acomodare pentru obişnuirea cu noile metode de lucru, orientate obiect; mai ales pentru programatorii ce au "crescut" cu limbajele de programare clasice. Dar este, sigur, mai simplu decît pare. Acest lucru ar vrea să-i demonstreze cele două listinguri care urmează.

Este de asemenea adevărat că cei odată "prinşi" greu renunţă la comodităţile obţinute. Programele sînt pur şi simplu mai transparente. Chiar dacă este necesară o anumită experienţă în lucrul cu clasele şi metodele, pentru a recunoaşte care funcţii pot fi formulate virtual, pentru evitarea codului inutil.

În cele două listinguri care urmează, EX72.CPP şi EX73.CPP metoda "Semn:: Indicare( )" este declarată ca funcţie virtuală. Pentru că afişarea unei linii sau unui dreptunghi se deosebeşte totuşi puţin de afişarea unui caracter individual, o primă diferenţă este deja numărul de atribute folosite pentru determinarea fără echivoc a obiectului definit.

Cele mai multe metode există pentru clasa de bază "Semn". Toate clasele derivate moştenesc aceste metode. Nicăieri nu este definită vreo metodă "Linie::Modificare( )" sau "Linie::Copiere( )". Nici nu este necesar. Cu ajutorul funcţiilor virtuale trebuiesc descrise numai excepţiile specifice claselor, şi deja există toate metodele corecte pentru urmaşi. Că acest lucru funcţionează, o ilustrează programul principal. Acolo aceleaşi metode se aplică claselor înrudite "Linie" şi "Dreptunghi". Nu ar trebui să existe reţineri în definirea şi testarea unor metode noi.

Modul de lucru al compilatorului Borland C++ 2.0 simplifică mult acest lucru. El oferă un debugger integrat, care sprijină mult parcurgerea pe paşi a programului. Programul EX72.CPP ar trebui neapărat parcurs în modul trace (), pentru a vedea efectul metodelor moştenite şi felul în care se "sare" la funcţiile specifice clasei. Desigur, ceva asemănător este adevărat şi pentru compilatorul Zortech. Între timp există şi un compilator shareware pentru C++, cu care se pot face primele experienţe.

```
// Exemplul 7.2
// metode pentru obiecte pluriforme

#include <conio.h>
#include <stdio.h>
#include <stdlib.h>

void Expand (char w, char a, int l, char *t)
{
 int i;

 for (i=0; i<=2*l; i=i+2) {
  t[i] = w; t[i+1] = a;
 }
}

class Semn  // defineste clasa Semn
{
 int x, y; // coordonatele

 public:
 char valoare; // valoare alfanumerica
 char attr; // atribut ecran
 Semn::Semn (int InitX, int InitY)
 {
  x = InitX;
  y = InitY;
 }
 int Semn::GetX ( ) {return x;}
 int Semn::GetY ( ) {return y;}
 void Semn::Position (int PosX, int PosY)
 {
  x = PosX; y = PosY;
 }
 void Semn::Citire( )
 {
  gettext (x, y, x, y, &valoare);
 }
 virtual void Semn::Indicare ( )
 {
  puttext (x, y, x, y, &valoare);
 }
 void Semn::Scriere (char Sign)
 {
  valoare = Sign;
  Indicare ( );
 }
 void Semn::Modificare (char Sign)
 {
  valoare = Sign;
  Indicare ( );
 }
 void Semn::BackColor ( )
 {
  int Color;

  if (attr > 128) {
   Color = attr - 128;
   Color = _rotr (Color, 4);
   attr = Color + 16 * Color + BLINK;
  }
  else {
   attr = _rotr (attr, 4);
   attr = attr + 16 * attr;
  }
 }
 void Semn::Stergere ( )
 {
  char Mvaloare, Mattr;
  // dupa stergere, in obiect ramin nemodificate
  // datele pentru valoare si atribut !!!

  Mvaloare = valoare; Mattr = attr;
  BackColor ( );
  valoare = ' ';
  Indicare ( );
  valoare = Mvaloare; attr = Mattr;
 }
 void Semn::Miscare (int NewX, int NewY)
 {
  Stergere ( );
  Position (NewX, NewY);
  Indicare ( );
 }
 void Semn::Copiere (int NewX, int NewY)
 {
  Position (NewX, NewY);
  Indicare ( );
 }
 void Semn::Set_attr (char fattr, char battr = 0,,\
           char Blink = 0)
 {
  if (battr < 8)
   attr = fattr + 16 * battr + Blink;
  else
   attr = fattr + Blink;
 }
};

class Linie : public Semn
{
 protected:
 int lungime;
 public:
 Linie::Linie (int InitX, int InitY)\
    : Semn (InitX, InitY)
 {
  lungime = 0;
 }
 int Linie::GetL ( ) { return lungime; }
 void Linie::Indicare ( )
```

```
112  {
113   char temp[4096];
114   Expand (valoare, attr, lungime, temp);
115   puttext (GetX( ), GetY( ),,\
116         GetX( )+lungime, GetY( ), &temp);
117  }
118  void Linie::Fix_lungime (int L)
119  {
120   lungime = L;
121  }
122 };
123
124 class Dreptunghi : public Semn
125 {
126  int LungX;
127  int LungY;
128
129  public:
130  Dreptunghi::Dreptunghi (int InitX, int InitY)\
131        : Semn (InitX, InitY)
132  {
133   LungX = 0;
134   LungY = 0;
135  }
136  void Dreptunghi::Indicare ( )
137  {
138   char temp [4096];
139
140   Expand (valoare, attr, LungX*LungY, temp);
141   puttext (GetX( ), GetY( ),,\
142         GetX( )+LungX-1, GetY( )+LungY-1, &temp);
143  }
144  void Dreptunghi::Fix_lungime (int lx, int ly)
145  {
146   LungX = lx;
147   LungY = ly;
148  }
149 };
150
151 int main( )
152 {
153  int x1, y1, s, z;
154  // fond cu semn albastru
155  x1 =1; y1 = 1;
156  Semn BSemn (x1, y1);
157  BSemn.Scriere (219);
158  BSemn.Set_attr (BLUE);
159  for (s=0; s<25; s++)
160   for (z=0; z<80; z++)
161    BSemn.Copiere (x1+z, y1+s);
162  printf ("Apasati o tasta ...");
163  getch ( );
164  // Reprezinta si apoi modifica o linie
165  x1 = 10; y1 = 10;
166  Linie LSemn (x1, y1);
167  LSemn.Set_attr (RED, BLUE);
168  LSemn.Fix_lungime (60);
169  LSemn.Scriere ('-');
170  getch ( );
171  LSemn.Miscare (x1+5,,y1+2);
172  getch ( );
173  LSemn.Stergere ( );
174  getch ( );
175  // reprezinta si apoi modifica un dreptunghi
176  x1 = 10; y1 = 2;
177  Dreptunghi DSemn (x1, y1);
178  DSemn.Set_attr (CYAN, BLUE);
179  DSemn.Fix_lungime (40, 20);
180  DSemn.Scriere (219);
181  getch ( );
182  DSemn.Miscare (x1+5, y1+2);
183  getch ( );
184  DSemn.Stergere ( );
185  getch ( );
186  clrscr ( );
187  return 0;
188 }
```

EX72.CPP

În listingul EX73.CPP attributul "lungime" a fost modificat în "Lungx", astfel încît pentru dreptunghi a mai trebuit declarat numai atributul suplimentar "Lung Y".

Cu noua funcţie element "Deplasare ( )" devine necesară modificarea la afişare a claselor "Linie" şi "Dreptunghi". Metoda "Deplasare( )" face posibilă modificarea la afişare a claselor "Linie" şi "Dreptunghi". Metoda "Deplasare( )" face posibilă modificarea poziţiei cu +/-1 pe direcţia x sau y. În contrast, metoda "Mişcare( )" transferă obiectul într-o poziţie nouă absolută.

În total există şase cazuri distincte de afişare a unei linii în raport cu fereastra ecran.

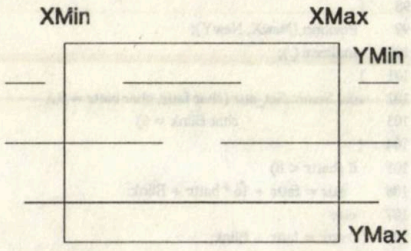


Fig. 1 Cazuri de pozitionare linii

Fie x1 punctul din stînga al unei linii, x2 punctul din dreapta (ele definind astfel segmentul reprezentat).

1) x1 x2 x Min
2) X Min x1 x2 X Max
3) X Max x1 x2
4) x1 X Min x2 X Max
5) X Min x1 X Max x2
6) x1 X Min X Max x2

Motivat de fereastra ecran, există linii care sînt afişate în întregime (cazul 2), linii "ciuntite" în stînga şi/sau în dreapta (cazurile 4,5 şi 6). Cazurile 1 şi 3 nu pot fi afişate, deoarece linia iese cu totul din fereastră. Pentru liniile afişabile, rezultă următoarea reprezentare.

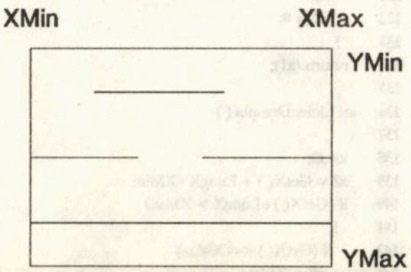


Fig. 2 Reprezentarea liniilor din fig.1

Aceleaşi cazuri există, prin analogie, la Dreptunghi.

La cele două exemple, în timpul ghidării pe ecran, nu se face nici o explicitare a tastelor cursor pentru deplasare, respectiv pentru dilatare sau comprimare. Dar pentru cei care-şi introduc singuri programele, aceasta nu

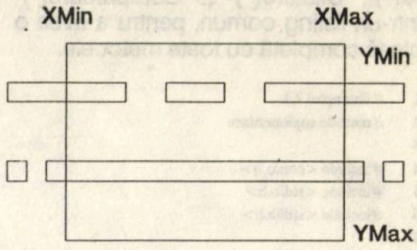


Fig. 3 Cazuri de pozitionare dreptunghiuri

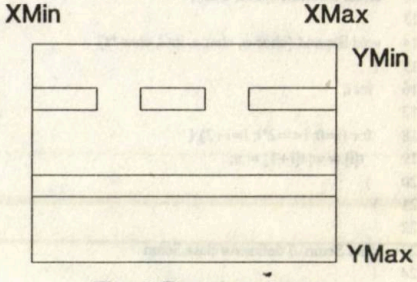


Fig. 4 Reprezentarea dreptunghiurilor din fig. 3

e o problemă. Pentru "Linie" a fost prevăzută dilatarea şi comprimarea pe direcţia x prin intermediul tastelor cursor - stînga şi respectiv cursor - dreapta. Celelalte două taste servesc la deplasarea în sus sau în jos.

Din motive de spaţiu s-a renunţat la deplasarea spre stînga sau dreapta. Desigur că nu va fi nici o dificultate pentru cei ce doresc să completeze.

La dreptunghi în schimb s-a implementat întîi de toate deplasarea în toate cele 4 direcţii. După terminarea deplasărilor (printr-o tastă "necursor") se trece în regimul dilatare - comprimare.

Problemele "laterale" ilustrate în fig. 1 şi 3 sînt rezolvate de funcţiile element "Stînga( )" şi Dreapta( )", ele returnînd punctele de început şi sfîrşit pentru porţiunea ce "se vede" în fereastră. Pentru "testul celor două funcţii "Dreptunghiul" ar trebui deplasat lateral pînă ce iese din cîmpul vizual, apoi în sus sau în jos şi apoi readus în fereastră.

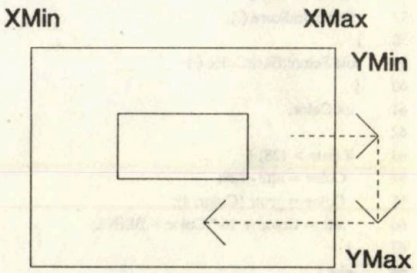


Fig.5 Propunere de test pentru EX73.CPP

După ultimele două listinguri, EX72.CPP şi EX73.CPP, nu va fi o problemă integrarea metodelor "Mişcare ()", "Copiere( )", "Ştergere( )", "Deplasa-

re( )", "Dilatare( )" şi "Comprimare( )" într-un listing comun, pentru a avea o clasă completă cu toate metodele.

```
// Exemplul 7.3
// metode suplimentare

#include <conio.h>
#include <stdio.h>
#include <stdlib.h>

const XMax = 80;
const YMax = 24;
const XMin = 1;
const YMin = 1;
enum Boolean {false, true};

void Expand (char w, char a, int l, char *t)
{
 int i;

 for (i=0; i<=2*l; i=i+2) {
  t[i] = w; t[i+1] = a;
 }
}

class Semn  // defineste clasa Semn
{
 int x, y; // coordonatele

 public:
 char valoare; // valoare alfanumerica
 char attr; // atribut ecran
 Semn::Semn (int InitX, int InitY)
 {
  x = InitX;
  y = InitY;
 }
 int Semn::GetX ( ) {return x;}
 int Semn::GetY ( ) {return y;}
 void Semn::Position (int PosX, int PosY)
 {
  x = PosX; y = PosY;
 }
 void Semn::Citire( )
 {
  gettext (x, y, x, y, &valoare);
 }
 virtual void Semn::Indicare ( )
 {
  puttext (x, y, x, y, &valoare);
 }
 void Semn::Scriere (char Sign)
 {
  valoare = Sign;
  Indicare ( );
 }
 void Semn::Modificare (char Sign)
 {
  valoare = Sign;
  Semn::Indicare ( );
 }
 void Semn::BackColor ( )
 {
  int Color;

  if (attr > 128) {
   Color = attr - 128;
   Color = _rotr (Color, 4);
   attr = Color + 16 * Color + BLINK;
  }
  else {
   attr = _rotr (attr, 4);
   attr = attr + 16 * attr;
  }
 }
 void Semn::Stergere ( )
 {
  char Mvaloare, Mattr;

  // dupa stergere, in obiect ramin nemodificate
  // datele pentru valoare si atribut !!!

  Mvaloare = valoare; Mattr = attr;
  BackColor ( );
  valoare = ' ';
  Indicare ( );
  valoare = Mvaloare; attr = Mattr;
 }
 void Semn::Miscare (int NewX, int NewY)
 {
  Stergere ( );
  Position (NewX, NewY);
  Indicare ( );
 }
 void Semn::Deplasare (int difx, int dify)
 {
  Stergere ( );
  Position (x+difx, y+dify);
  Indicare ( );
 }
 void Semn::Copiere (int NewX, int NewY)
 {
  Position (NewX, NewY);
  Indicare ( );
 }
 void Semn::Set_attr (char fattr, char battr = 0,\
                      char Blink = 0)
 {
  if (battr < 8)
   attr = fattr + 16 * battr + Blink;
  else
   attr = fattr + Blink;
 }
};

class Linie : public Semn
{
 protected:
 int LungX;
 public:
 Linie::Linie (int InitX, int InitY)\
    : Semn (InitX, InitY)
 {
  LungX = 0;
 }
 int Linie::GetL ( ) { return LungX; }
 int Linie::Stinga ( )
 {
  int x1;
  x1 = GetX( ) - XMin + 1;
  if (GetX( ) < XMin)
   {
    if (GetX( )+LungX > XMin)
     x1 = XMin;
    else
     x1 = 0;
   }
  return (x1);
 }
 int Linie::Dreapta ( )
 {
  int x2;
  x2 = GetX( ) + LungX - XMin;
  if (GetX( )+LungX > XMax)
   {
    if (GetX( ) <= XMax)
     x2 = XMax;
    else
     x2 = 0;
   }
  return (x2);
 }
 void Linie::Indicare ( )
 {
  char temp[4096];
  int x1, x2;

  x1 = Stinga( );
  x2 = Dreapta( );
  Expand (valoare, attr, x2-x1, temp);
  if (x1 != 0)
   if (x2 != 0)
    puttext (x1, GetY( ), x2, GetY( ), &temp);
 }
 void Linie::Fix_lungime (int L)
 {
  LungX = L;
 }
 void Linie::Dilatare ( )
 {
  Stergere ( );
  LungX += 1;
  Indicare ( );
 }
 void Linie::Comprimare ( )
 {
  Stergere ( );
  if (LungX > 0) LungX -= 1;
  Indicare ( );
 }
};

class Dreptunghi : public Linie
{
 int LungY;

 public:
 Dreptunghi::Dreptunghi (int InitX, int InitY)\
     : Linie (InitX, InitY)
 {
  LungX = 0;
  LungY = 0;
 }
 void Dreptunghi::Indicare ( )
 {
  char temp [4096];
  int x1, x2;

  x1 = Stinga( );
  x2 = Dreapta( );
  Expand (valoare, attr, (x2-x1+1)*LungY, temp);
  if (x1 != 0)
   if (x2 != 0)
    puttext (x1, GetY( ), x2,\
        GetY( )+LungY-YMin, &temp);
 }
 void Dreptunghi::Fix_lungime (int lx, int ly)
 {
  LungX = lx;
  LungY = ly;
 }
 void Dreptunghi::Dilatare (int x1, int y1)
 {
  Stergere ( );
  LungX += x1;
  LungY += y1;
  Indicare ( );
 }
 void Dreptunghi::Comprimare (int x1, int y1)
 {
  Stergere ( );
  if (LungX > x1) LungX = LungX - x1;
  if (LungY > y1) LungY = LungY - y1;
  Indicare( );
 }
};

int main( )
{
 int x1, y1, s, z;
 char tasta;
 Boolean sfirsit;
 // fond cu semn albastru
 x1 =1; y1 = 1;
 Semn BSemn (x1, y1);
 BSemn.Scriere (219);
 BSemn.Set_attr (BLUE);
```

```
234   for (s=0; s<25; s++)
235    for (z=0; z<80; z++)
236     BSemn.Copiere (x1+z, y1+s);
237   printf ("Apasati o tasta ...");
238   getch ( );
239   // Reprezinta si apoi modifica o linie
240   x1 = 10; y1 = 10;
241   Linie LSemn (x1, y1);
242   LSemn.Set_attr (RED, BLUE);
243   LSemn.Fix_lungime (60);
244   LSemn.Scriere ('-');
245   do
246   {
247
248    sfirsit = true;
249    tasta = getch ( );
250    if (tasta == 0)
251     {
252      sfirsit = false;
253      tasta = getch( );
254      switch (tasta)
255       {
256        case 72: LSemn.Deplasare (0, -1);
257              break;
258        case 80: LSemn.Deplasare (0, 1);
259              break;
260        case 75: LSemn.Comprimare ( );
261              break;
262        case 77: LSemn.Dilatare ( );
263              break;
264        // altceva decit tasta cursor
265        default: sfirsit = true;
266       };
267     };
268    } while (!sfirsit);
269   LSemn.Stergere ( );
270   getch ( );
271   // Reprezinta si apoi modifica un dreptunghi
272   x1 = 10; y1 = 2;
273   Dreptunghi DSemn (x1, y1);
274   DSemn.Set_attr (CYAN, BLUE);
275   DSemn.Fix_lungime (15, 10);
276   DSemn.Scriere (219);
277   do
278   {
279    sfirsit = true;
280    tasta = getch ( );
281    if (tasta == 0)
282     {
283      sfirsit = false;
284      tasta = getch ( );
285      switch (tasta)
286       {
287        case 72: DSemn.Deplasare (0, -1);
288              break;
289        case 80: DSemn.Deplasare (0, 1);
290              break;
291        case 75: DSemn.Deplasare (-1, 0);
292              break;
293        case 77: DSemn.Deplasare (1, 0);
294              break;
295        // nu e tasta cursor
296        default: sfirsit = true;
297       };
298     };
299    } while (!sfirsit);
300   DSemn.Scriere ('=');
301   do
302   {
303    sfirsit = true;
304    tasta = getch ( );
305    if (tasta == 0)
306     {
307      sfirsit = false;
308      tasta = getch ( );
309      switch (tasta)
310       {
311        case 72: DSemn.Comprimare (0, 1);
312              break;
313        case 80: DSemn.Dilatare (0, 1);
314              break;
315        case 75: DSemn.Comprimare (1, 0);
316              break;
317        case 77: DSemn.Dilatare (1, 0);
318              break;
319        // nu e tasta cursor
320        default: sfirsit = true;
321       };
322     };
323    } while (!sfirsit);
324   DSemn.Stergere ( );
325   getch ( );
326   clrscr ( );
327   return 0;
328  }
328
```

EX73.CPF

Cît ar fi durat elaborarea unui program, cu funcţiunile prezentate, într-un alt limbaj de programare, fără suportul orientării obiect?

Toate exemplele tratate pînă în momentul de faţă au în comun faptul că au lucrat exclusiv cu obiecte statice. Ceea ce înseamnă că necesarul de memorie e stabilit încă din momentul compilării şi este disponibil pe toată durata execuţiei. La obiecte mari, acest lucru poate să ducă la probleme cu memoria.

(Iosif Fettich)

La nivelul cel mai de jos

Turbo-Prolog partea a III-a

(continuare din pag. 45)

După ce ambele listinguri au fost compilate şi s-au obţinut fişierele obiect SUB.OBJ şi C_PRO.OBJ, pentru cazul nostru, comanda de editare a legăturilor va fi:

TLINK INIT C_PRO SUB C_PRO.SYM, CMIXED,, PROLOG.

În al doilea exemplu este cuplată cu un program Prolog tot o rutină C. În aceasta se lucrează cu numere reale, aşa că la editarea de legături vor fi necesare bibliotecile EMU.LIB, MATH.LIB şi CL.LIB. Modulul C calculează factorialul unei variabile întregi, rezultatul fiind de tip real (fig. 9).

După ce au fost create modulele obiect C_FAC.OBJ şi FACT.OBJ, cu comanda de editare a legăturilor: TLINK INLT FACT C_FAC FACT.SYM, FMIXED,, PROLOG + FMIXED.EXE.

```
global predicates
   sub (integer,integer,integer)
   - (i,i,o),(i,o,i),(o,i,i) language c

goal
   sub (100,150,X),
   write("100-150=",X,"\n"),
   sub (1000,Y,200),
   write("1000-",Y,"=200\n"),
   sub(Z,150,600),
   write (Z,"-150=600\n").
```

Fig. 7

```
global predicates
   factorial(integer,real) - (i,o) language c
goal
   write("Calculul factorialului cu ajutorul"),
   write(" unui modul extern scris in Turbo-C:\n"),
   write("Factorial de "),
   readint(N),
   factorial(N,X),
   write("Rezultat: ",X,"\n").

void factorial_0(int n, double *fac)
{
   double help;

   if (n<=0)
     {
       *fac=0.0;
       return;
     }

   for (help=2.0,*fac=1.0;n>1;n--,help+=1.0)
     *fac *= help;
}
```

Fig. 8

La utilizarea unor rutine Turbo-C mai trebuie să se ţină seama de:

1) cînd sînt apelate funcţii de bibliotecă în C acestea trebuie precedate de o linie de subliniere.
Deci "_test" în loc de "test". Acest lucru este necesar deoarece prin mascarea comutatorului compilatorului liniile de subliniere nu mai sînt generate în mod automat.

2) funcţiile apelate din Turbo-Prolog nu trebuie să returneze nici o valoare şi din această cauză trebuie definite cu "void".

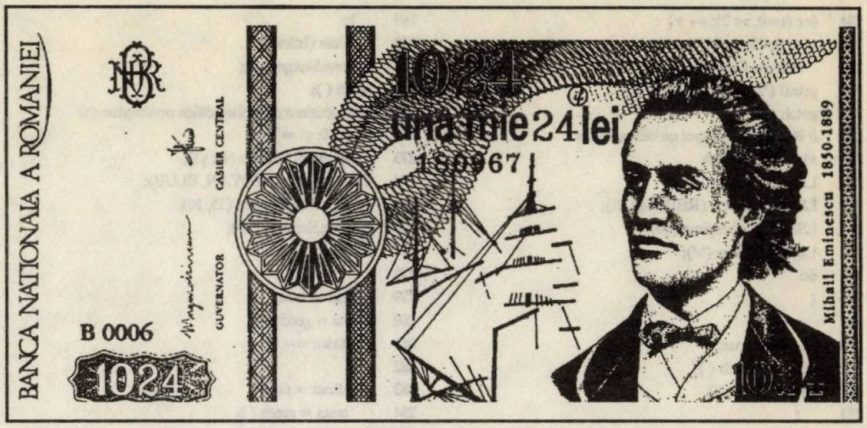
3) Funcţiile de ieşire "print" şi "putc" nu funcţionează atunci cînd Turbo-Prolog şi Turbo-C sînt utilizate împreună.

În numărul viitor vă vom prezenta modul de folosire a listelor în Prolog. Aşa cum s-a văzut Turbo-Prolog oferă posibilităţi de "programare logică". Programele date ca exemple au fost testate cu următoarele compilatoare: Turbo-Prolog 2.0, Turbo-Assembler 1.0 şi Turbo-C 1.5.

(R.M.)

1024

Conform promisiunii din numărul 7 al revistei, vom premia - în fiecare număr al revistei - cel mai interesant program **mai mic de 1024 octeţi** (în sursă) pe care îl primim spre publicare.
De data aceasta, sînt doi cîştigători, : **Vasile Brînduş şi Bogdan Munteanu** din Braşov. Felicitări !




```
1   program Park;
2
3   uses Dos,Crt;
4
5   var Regs : Registers;
6
7   procedure DPark (Drive : byte);
8    begin { DPark }
9    Regs.AH := 8;
10   Regs.DL := Drive;
11   intr ($13,Regs);   { obtinere parametri HD }
12   if Regs.AH = 0 then { drive-ul exista     }
13    begin
14     inc (Regs.CH);
15     if Regs.CH = 0 then { depasire }
16       Regs.CL := Regs.CL+$40;
17     Regs.AH := $0c ;
18     Regs.DL := Drive;
19     intr ($13,Regs);    { cautare pista }
20     write ('Hard-discul nr. ',chr(Drive-$80+ord('0')),' parcat ');
21     if Regs.AH = 0 then
22      writeln ('corect !')
23     else
24      writeln ('cu eroare ...',Regs.AH);
25    end;
26   end; { DPark }
27
28  begin  { Park }
29  ClrScr;
30  Regs.AH := 0;
31  Regs.DL := $80;
32  intr ($13,Regs);  { Resetare controller HD }
33  DPark ($80);      { Parcare primul HD     }
34  DPark ($81);      { Parcare al doliea HD  }
35  write ('Acum puteti opri calculatorul ...',^G^G);
36  inline ($FA/$F4); { blocare calculator cu cli / hlt }
37  repeat until false { bucla infinita }
38  end.  { Park }
```

PARK.PAS

Ce zi e astăzi ?

Programul ce urmează, scris în FORTRAN-77, tipăreşte data, ora şi ziua săptămînii pe display, pe imprimantă sau într-un fişier disc. Menţionez că ziua săptămînii se stabileşte automat, pe baza datei setate în sistem.

Sandu Răzvan,
Bv. Aerogării 12, 71547 Bucureşti I

Parcarea harddisk-urilor

Partea cea mai costisitoare şi, în acelaşi timp, cea mai sensibilă a unui PC este harddisk-ul. Este deci evidentă necesitatea acordării unei griji deosebite exploatării acestei componente.

O mare importanţă o are protejarea suprafeţei magnetice la transportul calculatorului şi la pornirea/oprirea lui. În ultimul caz, datorită fenomenelor electrice tranzitorii ce se petrec în aceste condiţii, este posibil ca prin capetele de scriere ale harddisk-ului să apară un curent electric accidental care să ducă la pierderea informaţiei înscrise anterior pe pista respectivă. Dacă ultima citire sau scriere s-a făcut pe pista ce cuprinde FAT-ul sau directorul rădăcină, atunci pagubele sînt considerabil mai mari.

Programul propus realizează deplasarea capetelor harddisk-ului dincolo de ultima pistă folosită şi blocarea calculatorului. În acest fel, la oprirea calculatorului capetele se vor găsi în dreptul unei piste nefolosite (deci care nu cuprinde informaţie).

ing. Vasile Brînduş,
str. Aurora nr. 25 bl. 10 ap. 32 2200 Braşov

ing. Bogdan Munteanu,
str. Operetei nr.52 C ap. 12, 2200 Braşov

```
1       SUBROUTINE WEEK(NL,N)
2   C   Aceasta subrutina scrie ziua saptaminii,,
3   C   data si ora pe unitatea logica NL,
4   C   aliniate cu N spaii
5       CHARACTER *9 SAPT(0:6),D,T
6       INTEGER ZI, ANLU(12), AN
7       DATA SAPT/'DUMINICA','LUNI','MARTI',
8      -'MIERCURI','JOI','VINERI','SIMBATA'/
9       DATA ANLU/0,3,3,6,1,4,6,2,5,0,3,5/
10      CALL IDATE(LUNA,ZI,AN)
11      CALL DATE(D)
12      CALL TIME(T)
13      A = A/4
14      M = INT(A)
15      J = M + AN + ANLU(LUNA) + ZI
16      K = MOD(J,7)
17      IF(N.NE.0) THEN
18      WRITE (NL,15) SAPT(K),D,T
19      ELSE
20      WRITE (NL,18) SAPT(K),D,T
21      ENDIF
22      RETURN
23  15   FORMAT (' ',<N>X,A9,2X,A9/' ',<N>X,A8)
24  18   FORMAT (' ',A9,2X,A9/' ',A8)
25      END
```

WEEK.F77

Program cu repetiţie

De multe ori am dori să executăm o comandă DOS de un număr de ori. Pentru aceasta ar trebui să tastăm de q ori respectiva comandă c. De exemplu, pentru a obţine 5 cópii la imprimantă a fişierului PRETURI.AUG ar trebui să folosim secvenţa:

PRINT PRETURI.AUG repetată de 5 ori, deci unde q=5 şi c=PRINT PRETURI.AUG.

O alternativă o oferă programul REPEAT.EXE al cărui apel trebuie să respecte sintaxa:

[d:][cale]REPEAT[.EXE] q c

unde d: şi cale reprezintă unitatea şi respectiv calea pînă la fişierul REPEAT.EXE, c este comanda DOS de executat, iar q este numărul de ori. Pentru cazul particular de mai sus vom folosi:

REPEAT 5 PRINT PRETURI.AUG

Iată acum sursa programului, compilată cu Turbo Pascal al firmei Borland.

stud. Socaciu Ioan - Tiberiu , Cluj

```
{$m 8192,0,0}

uses
 dos;

procedure help;
 begin
  writeln('Name    : REPEAT');
  writeln('Copyright: (C) 1991 SokySoft');
  writeln('Written  : by Socaciu Ioan - Tiberiu');
  writeln('Adress   : Ariesului 102/55, Cluj, Romania');
  writeln('Phone    : 95 - 140047    from Romania');
  writeln('Phone    : 40 - 5 - 140047 from abroad');
  writeln;
  writeln('[d:][path]REPEAT[.EXE]<n><dos_command>');
  writeln('repeat the dos command n times')
 end;

procedure comanda_dos(comanda:string);
 begin
  if comanda<>''
   then
    begin
     swapvectors;
     exec(getenv('comspec'),'/c '+comanda);
     swapvectors;
     if doserror<>0
      then
       writeln('comanda neexecutabila')
    end
 end;

var
 i,n:integer;

function valoare(s:string):integer;
 var
  i,code:integer;
 begin
  val(s,i,code);
  if code<>0
   then
    valoare:=0
   else
    valoare:=i
 end;

var
 a,s:string;

begin
 if paramcount>1
  then
   begin
    n:=valoare(paramstr(1));
    if n=0
     then
      help
     else
      begin
       s:='';
       for i:=2 to paramcount do
        s:=s+paramstr(i)+' ';
       for i:=1 to n do
        begin
         getdir(0,,a);
         writeln('<',i,'>',' ',a,'>',s);
         comanda_dos(s)
        end
      end
   end
  else help end.
```

REPT.PAS

Comenzi pentru subdirectoare

De multe ori cînd scriem un utilitar am dori ca acesta să poată avea o opţiune (/S de exemplu) pentru a continua respectiva munca şi pentru fisierele din subdirectoare. O alternativă o oferă şi programul SUBDIR scris în TurboPascal (al firmei Borland) şi care foloseşte procedurile FindFirst şi FindNext din unit-ul Dos. Nu aveţi decît de modificat procedura prel şi să setaţi corespunzător variabila sub.

stud. Socaciu Ioan - Tiberiu, Cluj

```
{SUBDIR.PAS}
{(C) 1991 SokySoft. Written by Socaciu Ioan - Tiberiu}
{TurboPascal is trademark of Borland}
uses
 crt,dos;
var
 sub:boolean;
procedure
 recursiv(path:string);
 var
  s:searchrec;
 procedure
  prel;
  begin
   writeln(path+s.name)
  end{begin};
 function
  director:boolean;
  begin
   with s do
   director:=(attr and directory=directory)and(name[1]<>'.')
   {endwith}
  end{begin};
 begin
  findfirst(path+'*.*',anyfile,s);
  while doserror=0 do
   begin
    if director and sub
     then
      recursiv(path+s.name+'\')
     else
      if not director
       then
        with s do
        if (name[1]<>'.') and (attr and volumeid<>volumeid)
          then
           prel
         {endif}
        {endwith}
       {endif}
     {endif};
    findnext(s)
   end{begin}
   {endwhile}
 end{begin};
begin
 clrscr;
 sub:=true;
 recursiv('d:\an2zi\')
end{begin}.
```

SUBDIR.PAS

Comenzi pentru subdirectoare

De multe ori cînd scriem un utilitar am dori ca acesta să poată avea o opţiune (/S de exemplu) pentru a continua respectiva munca şi pentru fisierele din subdirectoare. O alternativă o oferă şi programul SUBDIR scris în TurboPascal (al firmei Borland) şi care foloseşte procedurile FindFirst şi FindNext din unit-ul Dos. Nu aveţi decît de modificat procedura prel şi să setaţi corespunzător variabila sub.

stud. Socaciu Ioan - Tiberiu, Cluj


```
{SUBDIR.PAS}
{(C) 1991 SokySoft. Written by Socaciu Ioan - Tiberiu}
{TurboPascal is trademark of Borland}
uses
 crt,dos;
var
 sub:boolean;
procedure
 recursiv(path:string);
 var
  s:searchrec;
 procedure
  prel;
  begin
   writeln(path+s.name)
  end{begin};
 function
  director:boolean;
  begin
   with s do
   director:=(attr and directory=directory)and(name[1]<>'.')
   {endwith}
  end{begin};
 begin
  findfirst(path+'*.*',anyfile,s);
  while doserror=0 do
   begin
    if director and sub
     then
      recursiv(path+s.name+'\')
     else
      if not director
       then
        with s do
        if (name[1]<>'.') and (attr and volumeid<>volumeid)
          then
           prel
         {endif}
        {endwith}
       {endif}
     {endif};
    findnext(s)
   end{begin}
   {endwhile}
 end{begin};
begin
 clrscr;
 sub:=true;
 recursiv('d:\an2zi\')
end{begin}.
```


SUBDIR.PAS

Mica publicitate

Schimb programe Basic pentru compatibile Spectrum. Tel. 929/48591, Mihai

Vînd calculator "CIP", 64k, compatibil ZX-Spectrum, color. **Înregistrez** dischete 5,25" sau 2,5" cu diferite programe (JOCURI, UTILITARE, COMPILATOARE, SISTEME). Tel. 90/16.78.13

Am 15 ani şi învăţ la Liceul Teoretic "Avram Iancu" din Cluj, clasa a X-a, secţia de informatică...Aş dori ca prin intermediul revistei **să pot face cunoştinţă** cu tineri de vîrsta mea şi cu preocupări apropiate. Grama Radu, str. Gorunului, nr. 1, bl. D12, sc. B, ap. 20, 3400 Cluj Napoca.

REDUCERE !!

Începînd cu nr. 2/92 oferim tuturor celor interesaţi posibilitatea de a se abona la revista noastră la un preţ mai convenabil: oricine solicită un număr de cel puţin **10** exemplare, va avea o reducere de **10%**, dacă va achita în avans contravaloarea revistelor cerute (din care se scade 10% din valoarea lor). Deoarece preţul poate varia de la un număr la altul, vă rugăm fie să "deschideţi cont" la noi, fie să luaţi legătura cu redacţia.

Pentru cei care doresc numere mai vechi ale revistei: mai sînt disponibile o parte din revistele editate anul trecut, şi anume numerele 2-8/91. Acestea pot fi achiziţionate în aceleaşi condiţii ca mai sus.

Poşta redacţiei

De curînd, am intrat în posesia a două mouse-uri (unul de tip "Microsoft" iar celălalt de tip "Leonard") şi aş dori să ştiu ceva mai amănunţit cum funcţionează şi cum pot fi utilizate; de asemenea aş mai fi interesat şi de programele care fac posibilă utilizarea mouse-urilor. Dacă nu veţi avea suficient spaţiu, pentru a-mi satisface cerinţele, la rubrica "Poşta redacţiei", pe care o rezervaţi pentru corespondenţă, v-aş fi recunoscător dacă mi-aţi trimite datele solicitate (bineînţeles dacă este posibil) pe adresa:

Cartier Calea Bucureşti, Bloc B5, Scara 2, Etaj 3, Apartament 7, Tel. 45344, cod 1100, Localitatea Craiova, Judeţul Dolj.

Vă mulţumesc!

Blaghi Sebastian Paul

Cîteva informaţii veţi găsi poate în if nr.4/91 (7).

În revista dumneavoastră am găsit şi o serie de articole extrem de interesante despre programe ce pot prelucra şi interpreta imagini şi m-am gîndit ca împreună cu un profesor să fac un program ce să interpreteze amprentele digitale, iar în urma prelucrării imaginii furnizate să pună un diagnostic cît mai corect asupra bolii de care suferă pacientul. Am acces la un scanner cu ajutorul căruia să pot încărca imaginea, dar ceea ce îmi lipseşte este un program care să o analizeze. V-aş ruga, dacă puteţi, să-mi furnizaţi cîteva denumiri de programe care să realizeze asemenea lucruri, precum şi informaţii de unde le-aş putea procura. Aş mai dori să-mi recomandaţi şi bibliografie de citit despre programe şi despre domeniul respectiv.

Pentru a fi mai explicit aş dori un utilitar care să reuşească să recunoască şi să analizeze o dreaptă, o intersecţie de drepte, arce de cerc, de parabolă, bucle, etc. şi să-mi furnizeze coordonatele caracteristice fiecărui element al imaginii. Ştiu că pare cam mult ceea ce aş dori eu să realizeze un singur program, dar poate că reuşesc să combin între ele acţiunile mai multor programe.

Aştept cu nerăbdare răspunsul dumneavoastră pe adresa: Bucureşti, str. Ştefan cel Mare, nr. 2, bl. 13, sc. A, ap. 16, sect. 1, Tel: 11.09.28.

Cu mulţumiri,

Ionuţ Ciubotaru,
student medicină anul III

Pe toţi corespondenţii noştri îi rugăm să se uite şi peste talonul din acest număr şi serviciul Share-if pe care îl lansăm. Îi rugăm pe toţi cei care ne trimit materiale spre publicare să ne specifice în scrisoarea însoţitoare dacă sînt sau nu de acord ca materialele trimise să intre în baza pentru share-if, după, înainte sau în loc de publicarea în "if".

Iosif Fettich

Important !

Doriţi să aveţi garanţia că nu veţi pierde nici un număr din "**if**" ?
Încheiaţi un abonament folosindu-vă de talonul alăturat, sau adresaţi-vă difuzorilor noştri autorizaţi:

Alba Iulia
» Computer 5 S.R.L.
tel. 22173
Arad
» BB Computer
Boros Ladislau
tel. 17583, 31778
Bacău
» Pătru Liviu
tel. 44242
Baia Mare
» Riţiu Traian
tel. 25230
» Omni Computing
tel. 11794
» SINTEC S.R.L.
tel. 15720 - 171
Braşov
» Pummi S.R.L.
tel. 22018
Taifun S.R.L.
tel. 87368
» West Side Computers
tel. 49151
Brăila
» Teodorescu Constantin
tel. 47203
Bucureşti
» Dracea Tudor
tel. 210340
» Bronislaw
Telecom S.R.L.
tel. 382136, după ora 17
Buzău
» S.S.I. Infobuz -
Buzău S.A.
ing. Ursu Ioan
tel. 31844
Cluj Napoca
» S.C. "AbMod" S.N.C.
—Magazin
Hardware-Software
b-dul 22 Decembrie
nr. 135
—Sediul central
tel. 111090
» Costin Suciu Libris
tel. 145526
» Roşca S.N.C.
tel. 138773
Constanţa
» Ionescu Niculae
tel. 655077
Craiova
» Stătescu Paul
tel. 48117
Deva
» Toma Tudor
tel. 10144, 10234
Drobeta Turnu-Severin
» Bazavan Constantin
tel. 23666 - 143
Iaşi
» Buleu Daniel
Complex stud.
T. Vladimirescu
cam. T13 cam. 106
» KEYSYS S.R.L.
tel. 44499
Oneşti
» Brumă Cornel
tel. 24222-236
Oradea
» S.C. "AbMod" S.N.C.
tel. 60278, 44476
Petroşani
» S.C. Infotel-Pub-
Service S.R.L.
tel. 43100, 45537
Piteşti
» Sorel S.R.L.
tel. 23600
Rădăuţi
» Burciu Florin
tel. 63483
Satu Mare
» Viorel Santoi
Micro 17, str. Aniversării
bl. VA1, ap.6
Sibiu
» Pop Ioan
tel. 32011 / 208
Sighişoara
» Beer Mihai
tel. 73638
Suceava
» S.S.I. Suceava
Cazan Maria
tel. 15133
Timişoara
» Tomoroga Mircea
tel. 77422

Căutăm difuzori/distribuitori autorizaţi pentru difuzarea revistei "**if**" !. Condiţii avantajoase !
Rugăm persoanele interesate să ia legătura cu redacţia.

Revista "**if**" vă oferă spaţiu pentru publicitate şi reclamă la următoarele tarife:
o pagină alb-negru 50.000 lei
Pentru anunţurile care ocupă numai o parte din suprafaţa paginii, preţul este de 100 lei/cm^2

Un nou serviciu pentru cititorii noştri:

SHARE - if

Pentru o mai bună circulaţie a informaţiei între ofertanţi şi solicitanţi de programe (şi nu numai ...), vă oferim posibilitatea să:

- economisiţi timpul necesar introducerii programelor mai lungi, al căror listing a apărut în "if".
- preluaţi pe dischetă, pentru eventuale multiplicări, textul articolele din "if" pe care aţi vrea să le aveţi disponibile şi în această formă
- preluaţi acele programe şi texte de articole care ne-au fost trimise spre publicare, dar pe care, din variate motive (cel mai adesea, lipsa de spaţiu) - nu ajungem să le publicăm. (Desigur, aceste materiale le difuzăm numai atunci cînd există acordul autorului - încă nu am găsit o variantă de recompensare a autorului pentru această variantă de difuzare) O listă incompletă: criptare fişiere, completare formulare, o parolă solidă, nucleu de multitasking, "catalog" de dischete, etc.
- faceţi cunoştinţă cu acele programe din care ni s-au trimis demo-uri, respectiv **să difuzaţi programele dumneavoastră demonstrative** prin intermedierea noastră. Acest din urmă punct sperăm să intereseze în mod deosebit pe toţi autorii noştri de programe, cărora le este destul de greu să-şi facă publicitatea de care au, evident, nevoie.

Cei dornici să includem demo-urile lor în oferta noastră sînt rugaţi să achite, odată cu expedierea dischetei sursă, suma de 500 lei în contul nostru: Cont nr. 40729-96010402, Banca Română pt. Dezvoltare Mureş.
(Cei care trimit materiale informativ/didactice, şi nu publicitare, nu au de achitat nimic.) Preţul pe care îl practicăm pentru o dischetă este, din păcate, greu de definit în avans, din motive cunoscute de toată lumea. De aceea vă rugăm să vă interesaţi - telefonic sau în scris - în momentul lansării comenzii. În principiu, preţurile sînt identice cu preţurile cu care, la rîndul nostru, am achiziţionat dischetele, la care se adaugă 50 lei pentru munca noastră şi, suplimentar, taxa de expediere. Desigur, cei care ne trimit dischetele lor, au de achitat doar taxa de copiere şi expediere, iar cei care trec sau trimit pe cineva pe la redacţie, vor avea de achitat doar cei 50 lei, taxa de copiere.

Informaţii la zi din lumea calculatoarelor personale puteţi obţine numai citind regulat revista "if" !

Micro ATCI
C.P. 64, O.P. 1
RO - 4300 Tîrgu Mureş

Aveţi ceva de vîndut? Doriţi să cumpăraţi ceva ? Vreţi să vă oferiţi serviciile sau aveţi nevoie de ajutor într-o problemă ? Folosiţi mica publicitate de specialitate din "if" !

Micro ATCI
C.P. 172, O.P. 1
RO - 4300 Tîrgu Mureş

Talon de comandă-abonament

Subsemnat(a/ul).....
doresc să-mi trimiteţi |_____| exemplare din fiecare număr al revistei **if**, începînd cu numărul |_____| , pe adresa:
.....
.....
.....
Telefon: acasă..... la servici.....
Pentru aceasta am expediat prin mandat poştal/virament/cec.....lei în contul nr. 40729-96010402 deschis la Banca Română pt. Dezvoltare Mureş.

Abonamentul funcţionează pînă la epuizarea sumei; în cazul că banii pe care i-am trimis nu mai ajung pentru un alt număr, voi fi anunţat de acest lucru în timp util, astfel încît nu risc să pierd nici un număr. Dacă mă răzgîndesc şi renunţ la abonament, mi se returnează toată suma rămasă.

if 1 / 92 **Data: ..... Semnătura .....**

Talon pentru mica publicitate

Vă rog să inseraţi în numărul viitor al revistei **if** următorul anunţ:

|_|_|_

MicroInf - software profesionist pentru neprofesionişti!

CERBER
MICROLINK
μ UTILITY
FACT
CONFIG
MIDAS

- protecţia programelor şi datelor împotriva viruşilor şi accesului persoanelor neautorizate
- comunicarea între sucursale în mod rapid şi eficient
- confort în întreţinerea fişierelor
- o informare rapidă asupra tranzacţiilor efectuate
- configurator rapid şi flexibil
- achiziţionarea de date din procesul de producţie

Distribuitor autorizat: MBL Computers SRL
Bucureşti, Calea Dorobanţilor 152, sector 1, tel.333227,333706

μmicro ATCI S.R.L.

str. Gheorghe Doja 36
Tîrgu-Mureş ROMÂNIA
Adresa poştală: C.P. 64, O.P. 1
4300 Tîrgu-Mureş
ROMÂNIA
Telefon/Telefax: 004-954-31660

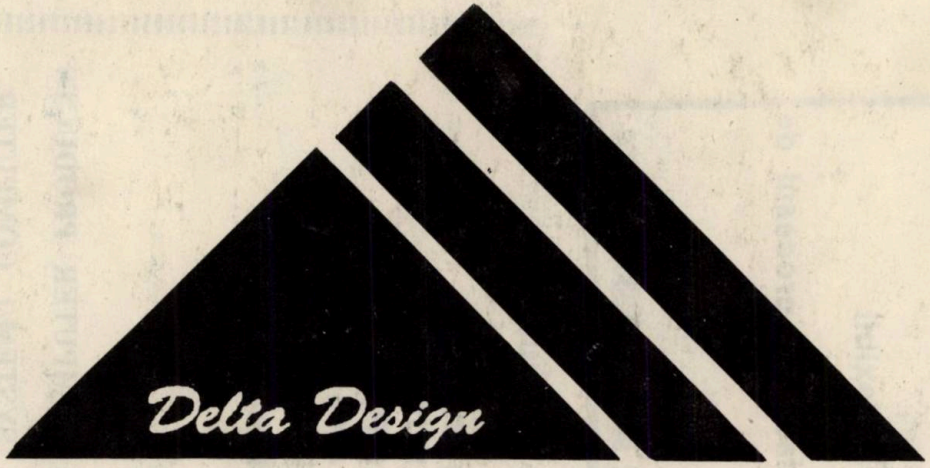
AUTHORIZED REMARKETER - IBM PERSONAL COMPUTER PRODUCTS
AUTHORIZED REMARKETER - IBM PERSONAL SYSTEM/1 COMPUTER

vă oferă

CALCULATOARE PERSONALE

din gama PS/1 şi PS/2

Raportul preţ/performanţă este în favoarea Dvs. !



Delta Design S.A., Hotel Bucureşti, 63-81 Victoria Street,
Wing D - Agencies, Stair E 2, 4th Floor, Sect. 1, Bucharest,
Phones: 15.42.26 / 40 015.45.80 Ext.2404 / Fax: 40 13.60.40.

99.94% of the users fully satisfied of the delivered products

---

Authorised distributors for the following companies:

Tulip Computers, Apple Computer Inc., Microsoft, Autodesk,
Ashton Tate, Claris, Corel, Delta Design International etc.

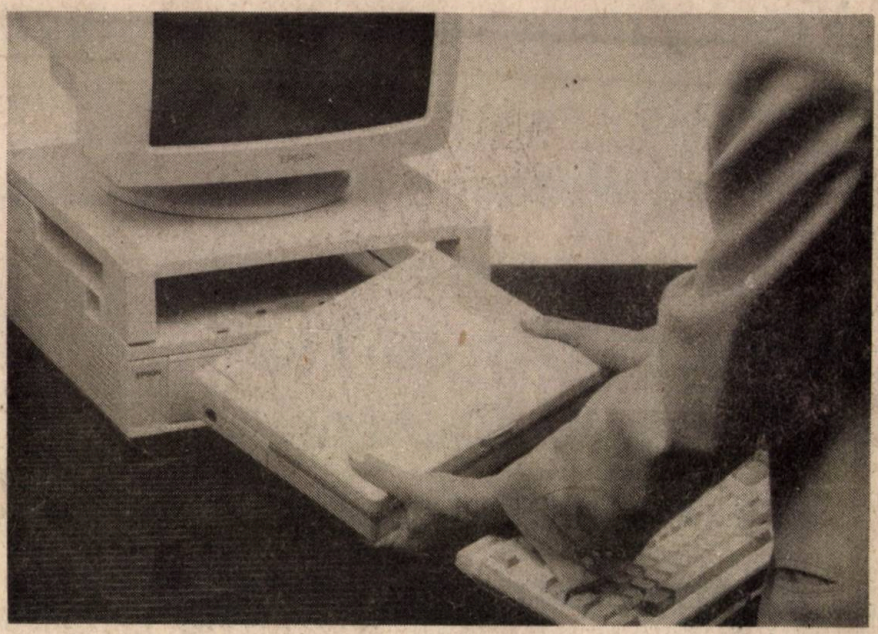
CALCULATOARE PERSONALE

Revistă editată de Micro ATCI S.R.L. Tîrgu Mureş

2 / 92
(13)
100 Lei
ISSN 1220-1529

Cuprins

Editorial 3
Ştiri 4
Produsele anului 1991 6
Vă prezentăm o firmă 8
Advanced Computing Environment 13
Standardul ACE devine realitate 16
FOURTH SHIFT - sistem de conducere 17
PC-MOS platformă multiuser DOS 20
Desktop Publishing 24
Calculatoarele sînt capabile să vadă 28
PC-uri extensibile (1) 32
Medii optice şi magnetice pentru stocarea masivă a datelor. 34
Super-floppy 38
Anatomia fişierelor PCX 40
Folosirea listelor în limbajul Prolog 46
Dinamica şi construcţia claselor - Curs C++ 50
Cînd cade reţeaua 56
Şotii 58
Comenzi pentru grupuri de fişiere 58
Deprotejarea fişierelor sub DR-DOS 58
Paza bună 59
Uite mouse-ul şi cînd nu e. 60
Ştiaţi că... 60
Share-if 61
Revista presei 62



Firma Epson ne propune cîteva noutăţi interesante:

- o combinaţie notebook, desktop. PC-urile de astăzi sînt flexibile, notebook-ul putînd fiind introdus în unitatea de extensie (v. foto) va dispune de o tastatură şi de un monitor normal şi va fi convertit astfel într-o confortabilă staţie de lucru. Unitatea de extensie mai dispune de două slot-uri libere de 16 biţi şi de două unităţi de stocare de masă care măresc capacitatea sistemului.
- imprimante care pot fi agăţate într-un cui pe perete. Imprimanta Epson LQ-100 este o imprimantă matricială cu 24 de ace destinată lucrului "at home". Magazia de hîrtie a imprimantei poate conţine pînă la 50 de foi, ea dispunînd de două fonturi scalabile şi de 5 fonturi LQ. Imprimanta este foarte silenţioasă, poate lucra aşezată orizontal sau vertical şi chiar şi agăţată într-un cui pe perete, permiţînd astfel economisirea spaţiului pe masa de lucru.

Ultima ora!

S-a infiintat POLICOM S.A. , *societate comerciala multinationala cu capital privat, avind principal actionar Ab Mod* S.N.C

Noua firma va ofera un sortiment larg de produse :

- *Calculatoare personale compatibile* IBM
- *Imprimante matriciale*
- *Dispozitive grafice*
- *Floppy - dischete*
- *Produse software originale*
- *Copiatoare*
- *Truse si accesorii pentru intretinere*

Nu uitati !

*POLICOM*S.A. *un partener potential pentru afacerile dumneavoastra*

CLUJ : B-dul 22 Decembrie nr. 135 tel. 95-156350

if

revistă de informatică
editată de firma **Micro ATCI**

Director: ing. Dumitru Dunca

Redacţia:
ing. Iosif Fettich,
ing. Ingrid Maier,
ing. Romulus Maier,

Colaboratori externi:
mat. Caraiani Eugenia
Caraiani Mitica
ing. Cetin Denislam
mat. Cozac Nelu
ing. Drosescu Radu
ing. Duşa Silviu
Golopenţa Sorin
ing. Onţanu Dan
ing. Pantea Adrian
Paris Cristian
stud. Socaciu Ioan-Tiberiu

Tiparul: Tiporex S.R.L.

Tiraj: 7.500 ex.

Preţ: 100 lei

Manuscrise originale sau listinguri de programe sînt primite cu plăcere de redacţie, cu condiţia să nu fi fost publicate şi în altă parte. Prin expedierea unui manuscris pe adresa redacţiei, autorul consimte implicit la publicarea materialului său în cadrul revistei. Onorariul se negociază cu redacţia. Materialele nepublicate nu se înapoiază şi nu se reţin.

Revista noastră vă oferă spaţiu pentru reclamă şi publicitate. Pentru amănunte vă rugăm să luaţi legătura cu redacţia.

Cei care doresc să anexeze revistei pliante publicitare tipărite în regie proprie, sînt rugaţi şi ei să se adreseze redacţiei .

Adresa şi telefonul redacţiei:
Micro ATCI, redacţia "**if**",
RO-4300 Tîrgu Mureş,
C.P. 172, O.P. 1,
tel./fax 954/31660,
41417 sau 46885 (după ora 19)
telex 65354 (Intertours)

Hainele cele noi ale împăratului

"Minunaţi-vă, minunaţi-vă de hainele cele noi ale **if**-ului", sau cum se spune într-un anume spot publicitar, "într-o nouă prezentare, cu un gust sublim. **if** style!". Încercăm sub acest nou chip tipărit (nu cioplit) să vă prezentăm mai multă informaţie şi să vă ajutăm să economisiţi cît de puţin din bugetul familiei. Şi dacă n-o să-i apreciaţi strălucirea, sperăm că cel puţin nu veţi spune că if-ul "e gol puşcă". Vă asigurăm că nu e gol şi că-n pieptu-i de aramă bat şapte vieţi, ba şi mai multe!

Am dori să facem o precizare pentru abonaţii care ne-au trimis bani pentru 12 numere pe 1991. Dată fiind întîrzierea cu care a apărut numărul 8/91, ne aflăm în imposibilitatea de a recupera numerele 9, 10, 11, 12/91. Considerînd că nu ar avea nici un sens să tipărim în luna mai 1992 un număr care să poarte numărul 9, sau 10/91, am decis să renumerotăm seria. În contul banilor expediaţi pentru numere care ar fi trebuit să apară anul trecut vă vom expedia numere care o să apară în acest an. Dacă nu sînteţi de acord cu această procedură vă rugăm să ne comunicaţi acest lucru şi vă vom returna suma rămasă.

Ce aduce nou acest număr? Pe lîngă rubricile consacrate deja: ştiri, teste comparative, produse noi, cursuri, vă semnalăm o concentrare mai puternică spre piaţa românească de informatică. Ne dorim, şi încercăm, ca revista noastră să devină o punte de legătură între producătorii şi consumatorii de produse informatice, fără însă a o transforma într-un fel de "Cronică a Parlamentului". Din această cauză nu vom prezenta doar produse noi pe plan mondial ci şi produse nou apărute pe piaţa internă.

Luna mai va fi (este) o lună bogată în expoziţii şi tîrguri (şi) de informatică (ROMORGADATA, CERF '92, TIBCO '92). Vom încerca să fim şi noi prezenţi şi să vă comunicăm noutăţile atîta vreme cît încă mai sînt calde şi nu după luni de zile (cum din păcate ni s-a întîmplat după TIB '91). Ştim că, şi ştim că ştiţi că ştim că, "veşnicia s-a născut undeva la sat" şi că România este un sat mai mare, dar vom încerca să ne scuturăm de mantia levantină.

Am putea să vă mai spunem multe care încep cu "vom", dar mai bine nu...

Aşa că cititorule care te afli pretutindeni în ţara asta (de la Nistru pîn' la Tisa, şi mai departe), clauza revistei de informatică celei mai citite dă-ne-o nouă astăzi şi nu ne duce pe noi în ispita de-a deschide o consignaţie. Căci dacă aveţi nevoie de noi, atunci noi, ca şi premierul, n-avem voie să spunem că sîntem obosiţi, nici măcar metafizic, şi, pendulînd între metafizică şi revistă, stăm şi facem revista pînă cădem laţi. Chiar dacă "Şi caii se împuşcă, nu-i aşa?"

Romulus Maier

¤ **Kodak** - *Fotografii profesionale pe CD-ROM*

Photo-CD deschide noi dimensiuni în transmiterea fotografiilor permiţînd accesul electronic la fotografiile color clasice şi permiţînd o prelucrare electronică a acestora. Photo-CD are o capacitate de 600 MByte, permite stocarea a 100 de fotografii color şi prin intermediul unei unităţi CD-ROM-XA permite transferul acestor fotografii la un PC. Sistemul permite stocarea fotografiilor digitale în 5 formate, cu rezoluţii cuprinse între 128 X 192 şi 2048 X 3072 pixeli şi dispune de interfeţe spre toate sistemele de operare uzuale disponibile pe PC-uri: DOS, OS/2, Unix. Principalul domeniu în care se scontează că sistemul va fi utilizat este domeniul Desktop Publishing.

Kodak AG, Hedelfingerstr. 60, 7000 Stuttgart 60, tel. 0711/406-0

¤ **Sanyo** - *Unităţi CD-ROM externe, interne şi portabile*

Firma Sanyo oferă noi unităţi CD-ROM ca soluţii standalone, built-in sau portable. Unitatea externă "CD-ROM 3000 GSE" şi cea internă "CD-ROM 4015 GSE" se cuplează prin intermediul plăcii de interfaţă Sanyo-Interface-Board (ISA). Pentru modelele IBM/PS2 există unitatea "CD-ROM 3000 GME" cu arhitectură microchannel. Toate unităţile dispun de o ieşire audio internă şi de o mufă externă pentru conectarea unor căşti. Complet ca şi kit cu Interface-Board, Caddy, Microsoft-Extension-Driver şi dischetă demo Sanyo, modelele GSE costă 1820 DM, inclusiv VAT. Modelul GME costă 2400 DM. Transferul datelor la PC se realizează prin intermediul interfeţei paralele, ceea ce permite utilizarea lor şi pentru Laptop-uri sau Notebook-uri care nu disoun de cartelă de interfaţă.

Sanyo, Truderinger Str. 13, 8000 München 80, tel. 089/416040

¤ **Wordstar** - *Un nou "Wordstar for DOS"*

Clasicul procesor de texte Wordstar se prezintă în această primăvară cu o haină nouă: vechea suprafaţă utilizator a fost înlocuită cu un front-end SAA. Aceasta înseamnă că urmaşul lui Wordstar 6.0 permite atît operarea cu un mouse cît şi darea comenzilor prin combinaţii de taste (short-cuts). Operarea a fost simplificată, meniurile sînt mai clar structurate şi nu sînt imbricate pe mai mult de două nivele. În această versiune este disponibil atît un Macro-Recorder cît şi un limbaj de programare a macrourilor. Graficele pot fi importate şi exportate direct fără a mai fi necesar programul auxiliar "Inset". Noile funcţii import/export permit stocarea unui întreg document în format PCX, format care poate fi prelucrat în continuare de majoritatea cartelelor fax.

Wordstar International, Meglingerstr. 2, 8000 München, tel. 089/785800-0

¤ **Computer 2000** - *HP New Wawe 4.0*

Încă înainte ca Microsoft să disponibilizeze Windows 3.1, Hewlett-Packard a anunţat o versiune dedicată acestuia a extensiei orientate obiect New Wawe. Versiunea 4.0 suportă şi utilizează toate caracteristicile lui Windows 3.1, cum ar fi OLE şi DDE. În plus produsul a fost îmbunătăţit, reducîndu-se necesarul de spaţiu pe harddisk de la 12 la 8 MByte, modulul Desktop Organizer permite acum şi funcţia Drag-and-Drop din Windows şi permite utilizarea unor nume de fişiere de pînă la 32 de caractere. De asemenea a fost extins limbajul de programare a macrourilor.

Computer 2000, Baierbrunner Straße 31, 8000 München 70, tel. 089/78040-0

¤ **Merisel** - *Lantastic 4.1 şi Lantastic for Windows*

Noua versiune Lantastic 4.1 dispune de un Systemmanager îmbunătăţit şi de putere sporită. Patru funcţii se îngrijesc de sporirea vitezei şi eficienţei reţelelor locale Lantastic: un Netbios mai rapid, Ressourcen-Caching, Random-Acces-Caching şi Record-Locking. "Lantastic for Windows" este un nou utilitar de reţea. Programul permite instalarea Lantastic sub Windows 3.0 în mod Real, Standard şi Enhanced, dispune de o interfaţă Windows pentru programele "NET" şi "NETMGR" şi permite DDE (Dynamic Data Exchanges). Lantastic for Windows costă 502 DM (incl. VAT).

Merisel, Peter-Henlein-Str. 2, 8037 Olching, tel. 08142/4180.

¤ **Computer 2000** - *Corel Draw 3.0*

Corel Draw 3.0 se află în faza de beta-testare. Printre cele mai importante funcţii noi, după datele producătorului Corel Systems din Canada, se numără o serie de Charting-Tools care vor permite elaborarea unor grafice de prezentare, cîteva noi efecte speciale şi utilizarea posibilităţilor OLE ale lui Windows 3.1. De asemenea va fi posibilă o editare direct în modul Preview, funcţia Undo va lucra pe mai multe nivele şi - în mod asemănător ca la Micrografx Designer - va exista un System-Layer care, printre altele, va împiedica modificarea accidentală a unor obiecte deja terminate. Textele vor putea fi editate şi în cadrul unui grafic şi nu numai în interiorul casetei de dialog. Formatul maxim al unei pagini nu va mai fi limitat la 17 X 17 inch.

Computer 2000, Baierbrunner Straße 31, 8000 München 70, tel. 089/78040-0

¤ **Plug-In** - *PC-ul ca aparat pentru dictat*

Cartela "PC-83" pentru înregistrarea şi redarea sunetului transformă PC-ul într-un aparat pentru dictat, cu ajutorul căruia se pot prelua înregistrări sau ştiri vorbite, se pot reda sau se pot transmite prin intermediul unui modem. PC-83 poate fi programată pentru înregistrare, redare şi acces la date în orice limbaj de programare de nivel înalt. Sînt posibile şi legături cu pachetele software deja existente scrise in GW-Basic, Quick- Basic, Pascal, MS-C, dBase III+ şi Clipper. Rata de transfer a vorbirii (viteza) poate fi programată soft, fiind posibile valorile 8, 4 sau 2 KByte/sec. Cu cît rata de transfer este mai mare, calitatea sunetului este mai bună, dar se pot memora mai puţine date pe disc. La o viteză de 2 KB/sec. pe un harddisk de 40 MByte se pot memora înregistrări de pînă la 340 min., la 4 KB/sec. doar 170 min., iar la 8 KB/sec. valoarea se mai înjumătăţeşte o dată. O cartelă, inclusiv un microfon şi un difuzor, costă 340 DM.

Plug-In Electronic Versand GmbH, Postfach 3 45, 8031 Eichenau, tel. 08141/72293, fax 08141-8343.

¤ **Canon** - *Canon CLC 10 copiază, scanează şi imprimă color*

CLC 10 se numeşte o nouă combinaţie copiator color, scanner şi imprimantă color, prezentată de curînd de firma Canon. Aparatul este de mărimea unei imprimante laser obişnuite, calitatea de redare a culorilor este excelentă şi va costa sub 20.000 DM. Aparatul se conectează la un PC prin intermediul unei cartele SCSI şi permite scanarea unei imagini de format maxim DIN A4 cu o rezoluşie de 400 X 400 dpi, semnal de culoare pe 24 de biţi, prin aceaşi interfaţă putînd ajunge la aceaşi rezoluţie şi la imprimare. Aparatul lucrează după modelul CMYK, după principiul

Bubble-Jet (este deci o imprimantă cu jet de cerneală) cu patru capete de imprimare, fiecare conţinînd cîte 128 de duze. Pentru copiere aparatul combină componentele de scanare şi imprimare, nu trebuie neapărat să fie cuplat la un PC, şi poate atinge o viteză de copiere de cca. 90 sec./copie color.

Canon Deutschland GmbH, Hellersbergstraße 2-4, 4040 Neuss 1, tel. 02101/125-0

¤ **Borland** - *Quattro Pro 4.0*

Deşi anunţase pentru mijlocul acestui an "Quattro Pro for Windows", Borland iese pe piaţă cu o nouă versiune DOS a aceluiaşi program Quattro Pro 4.0. Ce aduce nou această versiune? Enumerăm cîteva dintre aceste noutăţi: listă de butoane, asemănător ca la Excel şi Lotus for Windows, funcţii de analiză în timpul prezentării grafice, compatibilitate 1-2-3 îmbunătăţită (WK3, Allways, Impress, etc.), funcţii reţea îmbunătăţite, spooler de imprimantă, Print-Manager, posibilităţi extinse de configurare şi ajustare, ajutor în analiza pentru construcţia optimă a foilor de calcul, handling îmbunătăţit al foii de calcul (inserare bloc, sortare după coloane şi linii, etc.), bibliotecă de funcţii (Add-in funcţii @).

Borland GmbH, Gautinger Straße 10, 8180 Starnberg, tel. 08151/2630.

¤ **Next** - *Sistemul de operare Next disponibil şi pentru PC-uri 486*

Next Computer, firma lui Steven P. Jobs, a prezentat sistemul de operare orientat obiect "Nextstep" 3.0, care acum este disponibil şi pentru PC-urile care se bazează pe procesorul Intel 80486. Pentru Nextstep există deja mai mult de 500 de aplicaţii care vor putea fi astfel portate fără probleme şi pe platforma Intel. "Nextstep/486", acesta fiind numele versiunii Intel, are nevoie de un PC 486 cu cel puţin 8 MByte memorie (preferabil 16 MByte), un harddisk de cel puţin 120 MByte şi mouse. Reprezentarea se face deocamdată doar în nuanţe de gri cu cartele Super-VGA. Pentru prezentarea în pînă la 16 culori, din motive de viteză, va fi oferit suport, deocamdată, doar pentru cartele grafice EISA. Nextstep/486 va fi disponibil în a doua jumătate a lui 1992 şi în SUA va costa 995$.

Next Computer Deutschland GmbH, Oskar-Messter-Str. 24, 8045 Ismaning b. München, tel. 089/9612083.

¤ **Robotron** - *Un nou start în Est*

În fostul RDG Robotron avea un renume comparabil cu IBM sau Intel. Acesta n-a ţinut însă decît pînă la unificare. Sub numele "Soemtron", uriaşul concern, restructurat, încearcă să asambleze PC-uri la preţuri avantajoase. Noile calculatoare Soemtron utilizează motherboard-uri taiwaneze 486SX, cu procesoare Intel 486SX, cu tactul de 25 MHz, dispun de 4 MByte de memorie instalată, memorie extensibilă pînă la 16 MByte, au un lector de dischete de 3,5" şi unul de 5,25", harddisk Western Digital de 80 MByte cu timpul de acces de 17 ms, monitor Samsung, cartelă grafică cu o rezoluţie de 1024 X 768 dpi în mod interlaced, iar cartela I/O dispune de două interfeţe seriale, o interfaţă paralelă şi un Game-Port.

¤ **DEC** - *Surpriză(?) plăcută*

După căderea "Cortinei de fier" şi ridicarea restricţiilor de export la calculatoare, producătorii occidentali au început să-şi îndrepte atenţia şi spre ţările din estul Europei. Aşa cum se relata la sfîrşitul anului trecut în Wall Street Journal afacerile încheiate în 1991 sînt nesemnificative, dar potenţialul pentru viitor este ridicat. Mai ales Digital Equipment (DEC) a făcut surprinzătoarea descoperire că dispune de o clientelă numeroasă, cu toate că în urmă cu un an încă nu era activă în Europa de Est. Aceasta se explică prin faptul că în anii anteriori mai ales DEC a fost afectată de transferul ilegal de tehnologii. Datorită potenţialului ridicat al cîmpului aplicativ, în armată, industrie şi ştiinţă, al calculatoarelor VAX şi PDP- 11, acestea au fost foarte iubite în Est. Inginerii firmei cunoşteau acest lucru şi din această cauză pe cipurile aparatelor proprii aplicau inscripţia: "VAX. For those who care enough to steal the very best."

Ceea ce DEC nu putea şti însă era amploarea luată de această afacere "originală", recent au remarcat faptul că potenţialii clienţi cunoşteau bine de tot nu doar hardware-ul ci şi software-ul. Deşi clonele estice sînt mai proaste, asemănarea este suficient de mare pentru ca DEC să dispună de un vast cîmp de manevră. După părerea lui Pier Carlo Falotti, preşedintele pentru Europa al întreprinderii, mai ales cunoaşterea soft-ului va scuti DEC-ul de efortul unor ani îndelungaţi de şcolarizare. Aşa că pînă la urmă transferul de tehnologie neplătit din trecut se compensează pînă la urmă.

¤ **NEC** - *NEC colaborează cu CIC*

Communication Intelligence Corporation (CIC) şi NEC Japan au semnat o "Letter of intent" care permite firmei NEC să utilizeze, copieze şi să distribuie diferite produse CIC. Printre acestea se numără "Handwriter Multilingual Recognition Software", interfeţe spre diferite sisteme de operare şi "Handwriter On-Line Dynamic Signature Verification System". NEC doreşte să integreze aceste produse în sistemul de operare "Windows for Pen Computing" al firmei Microsoft pentru calculatoarele proprii.

¤ **Apple** - *Mărul discordiei*

Producătorul de calculatoare Apple şi casa de discuri Apple Records au depăşit scandalul de milioane care ia adus faţă în faţă pe băncile tribunalelor. Ambele întreprinderi utilizează ca logo un simbol asemănător. Logo-ul firmei Apple Computer este un măr cu dungi orizontale, cu o frunză în partea de sus, din care s-a muşcat o bucată, în timp ce logo-ul firmei Apple Records este un măr cu codiţă.

Agenţia de presă Associated Press (AP) a citat un avocat londonez care a mediat înţelegerea, dar amănuntele au rămas confidenţiale. Ziarul "San Francisco Chronicle" citînd o sursă anonimă a afirmat că Apple Computer a trebuit să plătească 29 de milioane de dolari.

Apple Corps, înfiinţată în 1963 de Beatles pentru a le proteja drepturile, a acuzat firma Apple Computer că i-ar fi provocat daune în 1981 deoarece mărul a apărut şi pe sintetizatoare muzicale. Controversa sa centrat mai ales pe MIDI-ul (Musical instrument digital interface) firmei Apple Computers. În cele din urmă Apple Records a permis producătorului de calculatoare să-şi aplice sigla pe calculatoare pentru a atenua posibilele confuzii în afacerile muzicale ale firmei britanice.

Produsele anului 1991

A devenit deja o tradiţie ca la fiecare început de an să se facă anchete în rîndul cititorilor pentru desemnarea celor mai bune produse ale anului anterior. Vă prezentăm în continuare cîteva topuri:

Clasamentul rezultat în urma răspunsurilor primite de la cititorii revistei **Computer persönlich** din Germania:

PC-uri 486	**Procent**
433 DE (Dell)	23,3
450 SE (Dell)	19,8
Highscreen 486 Tower (Vobis)	14,0
APC 486 (Dakota)	4,9
PC-uri 386	**Procent**
Highscreen Tower 386 DX-33 (Vobis)	21,1
APC 386 (Dakota)	18,3
Arco Balero (Tönnies)	10,1
System 325 (Dell)	6,5
PC-uri portabile	**Procent**
T3200 SXC (Toshiba)	16,7
T5200 (Toshiba)	7,8
LTE 386 (Compaq)	7,6
PAC 486 (Dolch)	7,5
Notebooks	**Procent**
320N (Dell)	14,7
NB386SX (Tandon)	13,2
LTE 386SX (Compaq)	7,7
Notemaster (Samsung)	6,5
Imprimante laser	**Procent**
Laserjet III (Hewlett-Packard)	57,8
Laserjet SI (Hewlett-Packard)	19,1
OL 800 (Oki)	3,1
HL 8 PS (Brother)	1,9
Monitoare	**Procent**
Multisync 4 D (NEC)	28,4
T 560 i (Eizo)	21,5
Flexscan 90 (Eizo)	13,6
Multisync 3D (NEC)	7,8
Procesare de texte	**Procent**
Word for Windows (Microsoft)	44,0
Word 5.5 (Microsoft)	18,6
Wordperfect 5.1 (Wordperfect)	9,4
Wordperfect for Windows (Wordperfect)	8,8
Calcul tabelar	**Procent**
Excel 3.0 (Microsoft)	56,2
Quattro 3.0 (Borland)	26,6
Lotus 1-2-3 3.1 (Lotus)	7,1
1-2-3 for Windows (Lotus)	2,1
Baze de date	**Procent**
dBase IV 1.1 (Borland-Ashton-Tate)	23,5
Superbase (SPC)	22,1
Paradox 3.5 (Borland)	16,3
Omnis Quartz (Blythe Software)	9,0
Grafică	**Procent**
Corel Draw (Corel)	49,8
Designer (Micrografx)	22,5
Artline (Digital Research)	0,9
Grafică de prezentare	**Procent**
Harvard Graphics 3.0 (SPC)	50,8
Charisma (Micrografx)	14,1
Excel (Microsoft)	6,3
Powerpoint (Microsoft)	4,4
Limbaje de programare	**Procent**
Turbo Pascal 6.0 (Borland)	35,6
Turbo C++ (Borland)	22,5
Visual Basic (Microsoft)	13,7
Quick Basic (Microsoft)	4,0

Firma **J.D. Power and Associates** a chestionat 1.100 de firme şi mai mult de 2.000 de utilizatori, obţinînd următorul clasament:

Calculatoare personale
1. Apple
2. Dell
3. Compaq
4. AST
5. Toshiba
6. Hewlett-Packard

Imprimante
1. Apple
2. Hewlett-Packard
3. NEC
4. Okidata

Software
1. Borland
2. Lotus
3. Wordperfect
4. Microsoft

Un alt chestionar al revistei **Computer Persönlich** s-a referit la calitatea service-ului oferit. Regretînd că din motive de spaţiu nu vă putem reproduce integral rezultatele sondajului de opinie, extrem de interesant chiar dacă se referă la Germania reproducem doar cîteva clasamente rezultate. (Nota maximă este 1, nota minimă este 4).

PC-uri: suport Hotline	**Nota**
HP	1,77
Compaq	1,85
IBM	1,97
Victor	2,00
Tandon	2,21
Peacock	2,28
Schneider	2,41
Siemens/Nixdorf	2,42
Commodore	2,75
Schmitt/Escom	2,77
Amstrad	2,80
Atari	2,88
Vobis	3,00
PC-uri: reparaţii	**Nota**
HP	2,00
Peacock	2,00
IBM	2,14
Victor	2,29
Siemens/Nixdorf	2,32
Compaq	2,33
Schneider	2,42
Tandon	2,55
Schmitt/Escom	2,56
Atari	2,75
Vobis	2,89
Commodore	2,97
Amstrad	3,64
Imprimante: Hotline	**Nota**
Oki	1,62
Kyocera	1,78
HP	1,91
Fujitsu	1,95
Star	1,97
Epson	1,98
NEC	2,07
Brother	2,25
Citizen	2,83
IBM	3,11
Imprimante: reparaţii	**Nota**
Oki	1,63
Fujitsu	1,68
Star	1,71
Epson	1,98
HP	1,99
Kyocera	2,00
Brother	2,29
NEC	2,32
Citizen	2,92
IBM	3,30
Software: Hotline	**Nota**
Star Division	1,68
SPI	1,83
Borland	1,88
Wordperfect	1,89
Central Point Software	1,91
Nantucket	1,94
Microsoft	2,19
Symantec	2,30
Lotus	2,79
IBM	2,88
Digital Research	3,00
Ashton-Tate	3,26

THE deal

HARDWARE

SOFTWARE

CAD - CAM

DESKTOP PUBLISHING

Technical support

TRAINING

Consulting

in p.c. field

雅仕 A&C INTERNATIONAL S.A.

Tel 40-0-53.53.15.

Vă prezentăm o firmă

Schimbările peisajului socio-politic din România ultimilor ani au fost însoţite de schimbări economice şi de structură, mai mult sau mai puţin vizibile, mai mult sau mai puţin necesare unei dezvoltări normale, unele pornite să aducă profituri rapide dar îndoielnice, altele, mai puţine, e adevărat, dar avînd obiective mult mai generoase, cu strategii gîndite şi aplicate de întreprinzători cu talent şi profesionişti cu experienţă reală în economia de piaţă.

Un exemplu martor al celei de a doua categorii de schimbări este **A&C INTERNATIONAL**, creată şi orientată într-o zonă extrem de fierbinte pentru orice societate - informatica şi informatizarea.

Ne aflăm astăzi la sediul firmei din Bucureşti, str. C. Tănase nr. 15.

Cristian Dumitrescu, preşedintele acestei firme, Ada Chan, vicepreşedinte, împreună cu Mihail Botez, directorul exècutiv, Daniel Vatcu, şeful departamentului service, Neculai Ghimbovschi, expert AutoCAD/AutoLISP, au avut amabilitatea să ne răspundă la cîteva întrebări. Vom încerca în acest fel să vă prezentăm nu numai echipa care asigură succesul acestei firme ci şi maniera firească dar obligatorie în care trebuie gîndită şi pusă la punct activitatea unei firme superspecializate cum este A&C.

Am optat pentru prezentarea sub formă de interviu a discuţiei noastre care, deşi mai puţin punctuală sau strict informativă, ne permite să vă redăm atmosfera absolut remarcabilă în care lucrează foarte tinerii angajaţi ai firmei.

Reporter: *Domnule preşedinte, din punctul meu de vedere, primul semn bun că avem de-a face cu o instituţie de nivel european şi nu doar balcanic, a fost vocea relaxată, binevoitoare şi cultivată a unei tinere care a răspuns la telefon: "A&C International, cu ce vă putem fi de folos?" Cum aţi defini d- voastră, în cîteva cuvinte, firma* **A&C INTERNATIONAL S.A.**?

Cristian Dumitrescu: A&C INTERNATIONAL este o firmă româno- canadiană cu o experienţă vastă în proiectare şi grafică asistată de calculator, în editare şi tehnoredactare computerizată şi procesare de imagini. Firma oferă servicii în design şi drafting, plotting, Desktop Publishing, printing, consulting în domeniul calculatoarelor şi training în toate domeniile enumerate mai înainte.

Mai mult, firma A&C are statut de Main Dealer pentru AutoDESK şi reprezintă în România, ca distribuitor şi dealer autorizat, multe firme de notorietate din lumea informaticii.

R: *Enumeraţi, vă rog, cîteva pentru cititorii noştri.*

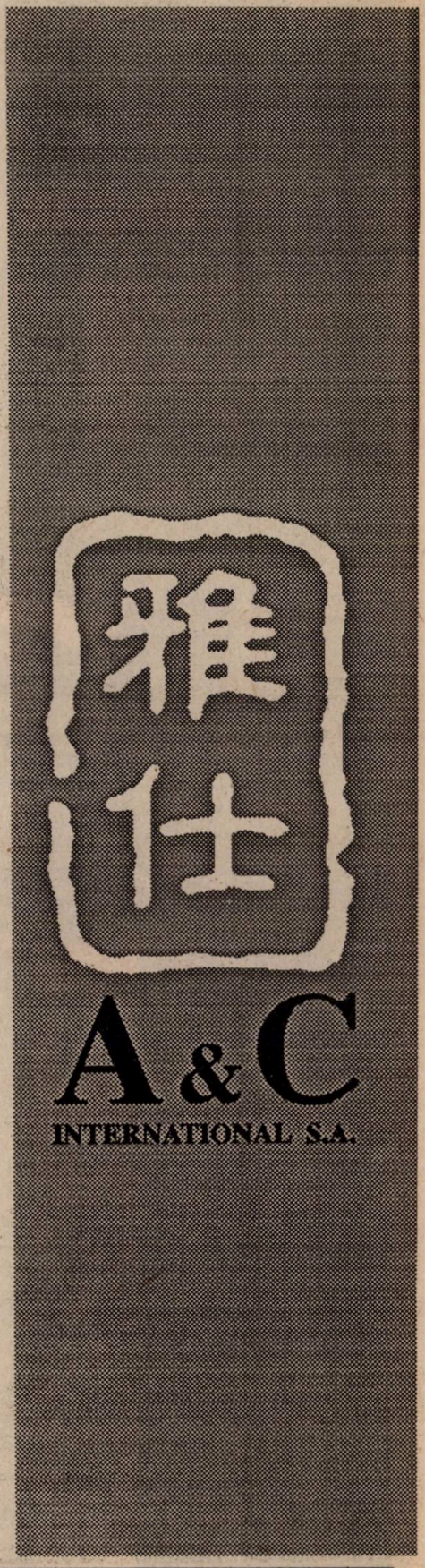
C.D.: Pentru cei care lucrează în informatică numele următoare vorbesc de la sine: AUTODESK, SYMANTEC, MICROSOFT, sau FOX, BORLAND, CALCOMP (o divizie a firmei LOCKHEED), PAGEMAKER, LOTUS şi altele.

R.: *În ce priveşte numele firmei, de ce A&C?*

C.D.: Simplu: Ada şi Cristian. Ada Chan este vicepreşedinte şi acţionar al acestei firme, ca şi în cazul firmei MBM-MAXIMUM BUSINESS MACHINES Inc. din Canada; aceasta din urmă produce computerele MAX în care fiecare componentă a fost gîndită şi finalizată avînd în vedere de la bun început utilizarea lor în proiectarea asistată şi în grafică. Realizate pentru marea performanţă, computerele MAX sînt comercializate numai prin firma A&C.

Pe Ada Chan, partenera mea aflată azi în Bucureşti pentru consultări pe probleme de management ale firmei, am cunoscut-o la firma HATCH la Toronto, unde lucra ca inginer constructor, şeful proiectului la care eu acordam consultanţă în proiectare cu AutoCAD. Absolventă a Institutului de Construcţii Civile (Civil Engineering) al Universităţii din Toronto, avînd un doctorat în Ştiinţe Aplicate (Master of Applied Sciences), Ada este un foarte bun specialist în probleme de rezistenţă a materialelor şi modelare cu element finit.

Împreună am pus bazele corporaţiei **MBM Inc.** din Canada, ale cărei produse le-am prezentat la TIB'90, TIB'91; sîntem de asemenea parteneri la firma **ATC INTERNATIONAL** din Hong Kong şi tot împreună cu Ada am înfiinţat anul trecut în România această firmă pe care



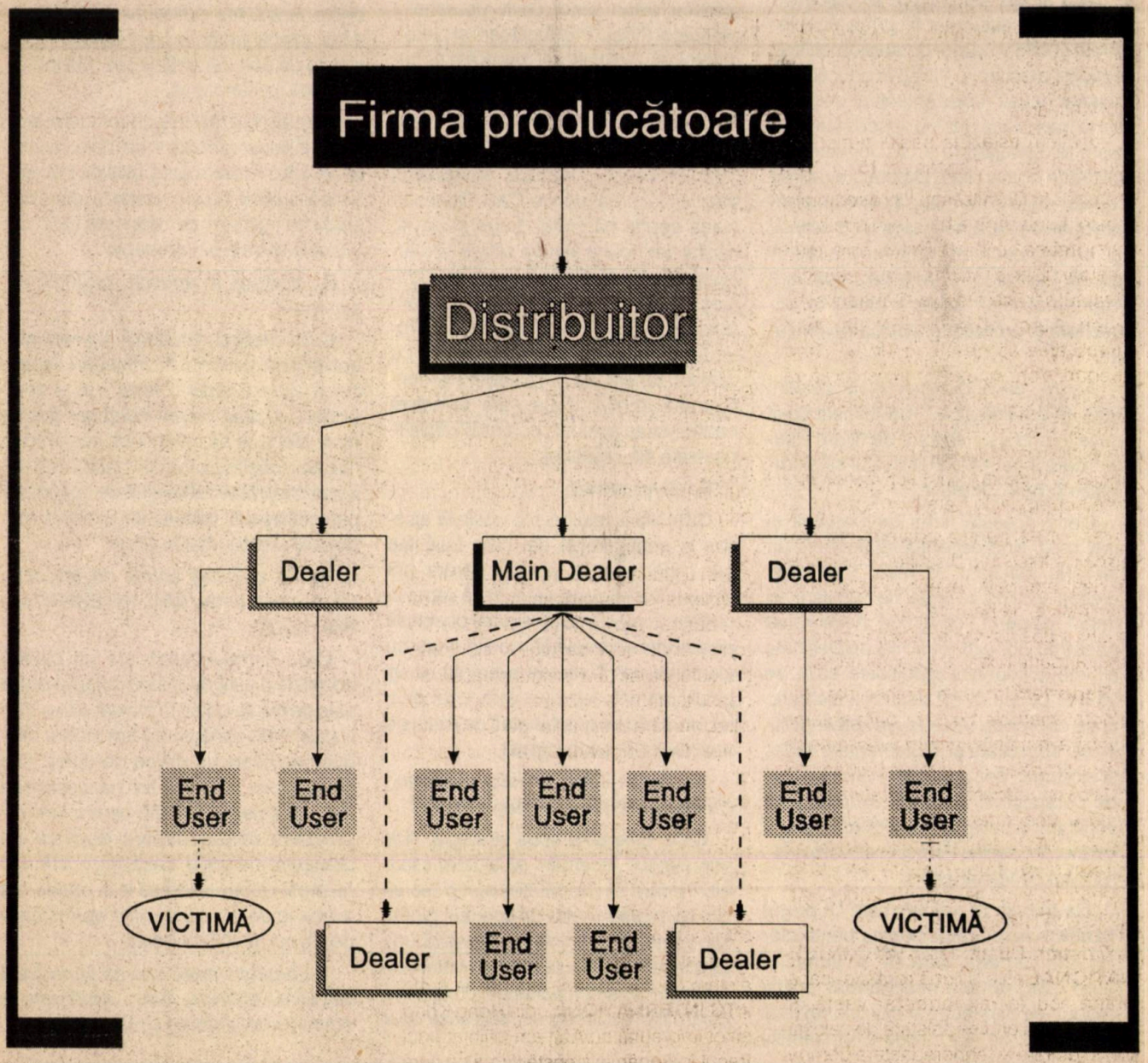
am numit-o **A&C INTERNATIONAL S.A.**

R.: *Este ora nouă seara şi văd că aproape toţi angajaţii firmei sînt încă în faţa calculatoarelor!*

C.D.: Sînt prietenii mei. Toţi sînt foarte tineri, pasionaţi de ceea ce fac, fiecare este foarte bun într-un anumit domeniu, dar sînt conştienţi că un profesionist se formează în timp şi cu multă muncă. Nu există obligaţii formale, rezultatul este singurul criteriu de apreciere şi ei ştiu că succesul lor depinde de succesul firmei.

R.: *Cum v-aţi format echipa? Există o organigramă în acest sens, cu un număr limitat de posturi?*

Daniel Vatcu: V-aş putea răspunde eu. În octombrie 1990 cînd MBM avea spaţiul său de expunere în standul canadian de la TIB, m-am oferit să asigur eu service-ul pentru calculatoarele acestei firme în cazul în care d-l Dumitrescu, reprezentantul MBM, s-ar fi arătat interesat. La data respectivă eram student în anul V al Facultăţii de Electronică din Bucureşti. Din momentul în care d-l Dumitrescu a decis să creeze firma româno-canadiană A&C, a hotărît să adopte şi să susţină în mod constant o politică de angajări foarte importantă pentru noi: folosind dotările performante ale firmei şi experienţa extraordinară în domeniu a d-sale, acumulată în cei 13 ani petrecuţi în Canada, a oferit unui număr de studenţi din Politehnică posibilitatea de a-şi finaliza proiectele de diplomă pe computere MAX în AutoCAD. Pe cei mai buni i-a angajat imediat după examenul de stat. Observaţi că timp de luni de zile firma nu a făcut nici un fel de afaceri, ocupîndu-se doar de formarea şi selecţia celor ce mai tîrziu aveau să formeze echipa! Dacă cineva din echipă are nevoie de încă un om în departamentul său, e liber să-l angajeze, dar va fi extrem de prudent, fiindcă



va împărţi cu acesta şi succesele şi lipsa eventuală de randament.

C.D.: Media de vîrstă a team-ului nostru este sub douăzeci şi opt de ani, deşi nu toţi sînt la începutul carierei. Firma beneficiază şi de experienţa în proiectare şi management a d-lui Ion Ismail, actualul director general şi a d-lui Mihail Botez, directorul nostru executiv.

Pentru postul de secretară am organizat o selecţie anunţată într-un jurnal de mare tiraj.

R.: *În activitatea dumneavoastră, cuvinte ca distribuitor, dealer, soft cu licenţă, sînt foarte obişnuite. Ne puteţi preciza semnificaţia lor?*

Ada Chan: Vă propun să discutăm pe schema alăturată: în urma unei instruiri sub forma unor cursuri speciale şi aplicaţii practice, o Firmă Producătoare selecţionează reprezentanţi ai unei alte firme care, dacă trec cu succes anumite teste de specialitate şi fac dovada unei eficienţe comerciale, capătă statutul de "dealer" care le permite să ofere pe piaţă produsele respective, asigurînd servicii şi consultanţă tehnică eventualilor clienţi (utilizatori). Firma producătoare impune acestor "dealer-i" criterii de eficacitate şi protocol specifice domeniului.

Acum, dacă din mai mulţi dealer-i unul se dovedeşte a fi mai eficient decît ceilalţi, primeşte statutul de "Main Dealer" care îi permite la rîndul lui să-şi formeze proprii "dealer-i".

Dacă această firmă se dezvoltă şi mai mult, fiind capabilă să acopere un teritoriu suficient de mare încît să-şi creeze în timp o reţea de "dealer-i" şi "main dealer-i", dovedind necesitatea penetrării mai extinse şi mai diversificate a pieţei produselor specificate, ea nu se va mai ocupa direct de comercializare, ci va coordona reţeaua, va stabili politica de dezvoltare şi implementare, tactic şi strategic, deci cu termen imediat sau de perspectivă, primind - în urma unor verificări extrem de minuţioase la faţa locului din partea Firmei Producătoare - statutul de "distribuitor".

Ce trebuie să reţineţi, este că **preţul fixat de producător pentru beneficiarul din capătul celălalt al schemei** (end user) **rămîne neschimbat**, oricîte blocuri de distribuţie ar conţine reţeaua pînă la el; se vor micşora procentele acordate intermediarilor (dar nu şi cîştigul lor, deoarece ei trec, să zicem, de la nivelul de "dealer" la cel de "main dealer", doar dacă numărul clienţilor lor a crescut foarte mult; deci, ceea ce pierde per unitate de produs, este mai mult decît compensat prin suma globală a cîştigului).

Clientul care cumpără direct de la unul din segmentele unei astfel de scheme, beneficiază de acea "licenţă" care îi asigură toate avantajele asistenţei tehnice prin specialiştii din reţea.

Dacă însă el, la rîndul lui, vinde obiectul procurat din reţea, (acţiune pedepsită de lege în alte ţări), noul cumpărător va fi păcălit, fiindu-i imposibil să primească vreun ajutor de la personalul calificat al Firmei Producătoare şi pierde dreptul de a obţine noi revizii ale softului, gratuite sau cu mici diferenţe de preţ.

R.: *În activitatea firmei A&C primează softul sau hardul?*

C.D.: Vedeţi d-voastră, computerul MAX a fost creat pentru CAD. Deci s-ar putea spune că softul "trage după el" hardul. Iar noi nu sîntem simpli comercianţi. Dacă beneficiarul ne spune: "doresc tipul X de computer", sigur că i l-am putea oferi fără alte discuţii. Dar noi procedăm altfel: stabilim împreună cu clientul, funcţie de necesităţile lui, configuraţia ideală, după care urmează confruntarea acesteia cu posibilităţile financiare ale clientului.

R.: *Şi rezultatul?*

C.D.: Rezultatul va fi o posibilă ajustare a configuraţiei din care însă fac parte elemente în privinţa cărora noi sfătuim beneficiarul să nu facă rabat la calitate; în plus, trebuie să lase deschisă posibilitatea completării configuraţiei cu elemente noi. Dacă de exemplu nu îşi poate permite achiziţionarea unui plotter, noi îi oferim posibilitatea de execuţie a plotărilor în cadrul firmei.

R.: *Intrăm astfel în zona serviciilor. Ce înseamnă ele în strategia firmei?*

C.D.: Este bine spus strategie. Noi, prin servicii, încercăm să ajutăm clientul. Fiecare din activităţile de tip "servicii", fie că este vorba de scanări, plotări sau asistenţă tehnică, se bazează una pe alta; ele constituie în mod evident un profit pentru ambii parteneri, chiar dacă pentru firmă aceasta se realizează doar pe termen lung. În general, am putea spune că deviza noastră este:

"Succesul d-voastră este măsura succesului nostru!"

R.: *Ce avantaje oferă o afacere încheiată prin firma d-voastră?*

C.D.: Pe lîngă ceea ce oferim conform statutului nostru profesional, există garanţiile oferite de numele firmelor de soft şi hard pe care le reprezentăm. Mai mult, unele firme producătoare oferă reduceri importante pentru educaţie-învăţămînt, pentru instituţii guvernamentale, ba chiar, la intervenţia noastră, preţuri avantajoase pentru România faţă de celelalte ţări din Est. Există şi reduceri de preţ funcţie de numărul de pachete AutoCAD solicitate. Prin firma A&C, clienţii noştri se pot abona la revistele de soft din străinătate, plătind în monedă românească.

Şi poate cel mai important lucru: noi avem în fiecare moment ultimele variante de soft din toate categoriile de interes, iar specialiştii noştri urmează periodic stagii de instruire, de "aducere la zi" cu ultimele noutăţi din domeniu.

R.: *Sprijiniţi în special învăţămîntul superior?*

C.D.: Desigur. Studenţii şi tinerii absolvenţi sînt receptivi, maleabili, entuziaşti, nu au căpătat acea rutină care oboseşte, deci putem colabora foarte bine. Mai ales că nu au încă acea mentalitate, specifică modului oriental de rezolvare a problemelor pînă de curînd, ce caracterizează cumva un procent important al celor mai în vîrstă.

R.: *Ce ne puteţi spune despre cifra de afaceri a firmei* **A&C INTERNATIONAL S.A.**?

C.D.: Firma noastră are un capital social de o sută de mii de dolari şi două sute de mii de dolari în mijloace fixe. Din august 1991 pînă acum am vîndut produse de peste un milion de dolari. Din cîştigul total, după ce am reinvestit cea mai mare parte în dotări pentru firmă şi programe de specializare, după ce am recuperat investiţia, ceea ce ne-a rămas ca profit efectiv se ridică la suma de opt milioane de lei, adică doar aproximativ douăzeci de mii de dolari.

R.: *Aceasta arată clar că firma* **A&C INTERNATIONAL S.A.** *urmăreşte o strategie de afirmare şi dezvoltare şi nu*

THE right option in hardware

MIX®
COMPUTERS
with
ATI Graphic Cards
MAG Monitors

PC MAGAZINE EDITOR'S CHOICE
1991 BYTE AWARD OF EXCELENCE

CalComp
Panasonic
HEWLETT PACKARD

雅仕 A&C INTERNATIONAL S.A.

Tel 40-0-53.53.15.

realizarea rapidă a unor profituri maxime dar fără perspectivă.

C.D.: Este foarte adevărat. Este una din primele idei de politică economică pe care le transmitem tinerilor noştri colaboratori. Sîntem deja douăzeci de persoane în echipă şi sînt foarte mulţumit că nimeni nu a fost nici dat afară şi nici nu şi-a dat demisia. Asta înseamnă pe de o parte că selecţia noastră a funcţionat perfect şi, pe de altă parte, că sîntem capabili să asigurăm stabilitatea firmei **A&C INTERNATIONAL S.A.** de la bun început.

R.: *Ce dificultăţi aţi întîmpinat în activitatea d-voastră?*

C.D.: În primul rînd ne-am lovit de cîteva carenţe legislative. Un exemplu ar fi plafonarea şi îngrădirea sponsorizării şi publicităţii. Alt exemplu, deosebit de important pentru noi, este legat de plăţile pe care le avem de făcut către firmele pe care le reprezentăm: legislaţia valutară actuală ne împiedică să respectăm termenele fixate pentru aceste plăţi.

R.: *Şi cum vă descurcaţi atunci?*

Mihail Botez: Norocul nostru - repet, norocul nostru! - este că avem în Canada firma MBM. Ea face plăţile pentru noi, încasînd banii ulterior; este, deci, un fel de "tampon"... Dar aş mai aminti aici şi de altă problemă, legată de valută: piedicile întîmpinate atunci cînd vrem să trimitem oamenii la specializare! Asta ca să nu mai vorbim de dificultăţile în obţinerea vizelor pentru specialiştii noştri care sînt aşteptaţi de un Training Center al unei firme, pentru a participa la cursuri!

R.: *De ce nu vă extindeţi activitatea în exportul unor produse ce ar putea prezenta interes? Nu ar reprezenta aceasta o posibilă soluţie a problemei plăţilor?*

C.D.: Noi sîntem profesionişti. Se ştie că nu te poţi menţine în topul specialităţii tale dacă foloseşti timpul pe care îl ai la dispoziţie pentru activităţi de altă natură decît cele pentru care eşti cel mai bun.

R.: *Am înţeles că activitatea firmei A&C din Bucureşti este continuarea activităţii d-voastră din Canada. Ne puteţi spune cîteva rezultate pe care le consideraţi semnificative?*

C.D.: Ne este foarte clar că în România ducem o muncă de pionierat şi ne bucurăm întotdeauna cînd oamenii realizează importanţa şi avantajul pătrunderii tehnologiei de vîrf în sectoarele lor de activitate. Prin calitatea asigurată de specialiştii noştri am cîştigat clienţi importanţi ca RENEL, Centrul Nuclear Cernavodă, AECL-ANSALDO Consortium, SNCF Română, COCA-COLA, GOLDSTAR, Electromagnetica, PetrolImport-Export, ELBA, Institutul de Fizică Atomică, ICEPRONAV, diferite edituri şi institute de proiectare şi foarte multe instituţii de învăţămînt superior, etc. În luna februarie 1992 am participat la expoziţia organizată de AGIR, în competiţie directă cu alte firme de profil. Am avut satisfacţia să vedem cum, la cîteva zile după închiderea expoziţiei, firma A&C a fost asaltată, am putea spune, de solicitatori şi comenzi pentru produsele promovate de noi.

R.: *Firma A&C manifestă un interes particular pentru CAD, pentru grafica asistată?*

M.B.: Este adevărat, noi punem accentul pe softul de grafică, deoarece este cel mai sofisticat, mai pretenţios, şi, zicem noi, cel mai interesant. Experienţa ne arată că dacă un echipament este performant pentru CAD, cu atît mai mult va putea fi utilizat sigur şi fără probleme pentru alte activităţi (organizare, contabilitate, etc.).

Noi considerăm grafica o elită a aplicaţiilor pe calculator.

Şi mai există un argument: toţi, începînd cu preşedintele firmei, sîntem nu numai specialişti, dar şi pasionaţi de acest domeniu.

R.: *V-aş întreba atunci, d-le Dumitrescu, de ce după atîţia ani de succes într-o ţară cu o civilizaţie tehnică avansată, aţi revenit în România înfiinţînd firma* **A&C INTERNATIONAL S.A.***?*

C.D.: Pur şi simplu fiindcă aici este casa mea.

Acum doi ani în România erau 3.000 de PC-uri; acum sînt peste 30.000. Pentru mine asta înseamnă ceva şi anume că o enormă carenţă a lăsat locul unei largi deschideri, unei mari disponibilităţi. Ne place să credem că A&C nu a ajutat doar la umplerea unui gol, dar a contribuit, prin activităţile sale specifice, la stabilirea unui anume nivel al calităţii de care va trebui să se ţină seama.

R.: *O întrebare pentru colaboratorii d-voastră: de ce lucraţi la A&C şi nu la o altă firmă?*

Neculai Ghimbovschi: Toţi cei de aici avem în comun pasiunea pentru CAD. Pot spune fără nici un fel de reticenţe că fac exact ce-mi place mai mult. Şi asta în condiţii de dotare şi atmosferă de lucru excepţionale.

M.B.: Preşedintele firmei, ba chiar şi noi, am făcut destule sacrificii pentru a o putea înfiinţa; ce m-a făcut să rămîn alături de el a fost încrederea pe care mi-a acordat-o.

Pe de altă parte, poate că ar fi bine să se ştie că, investind în alte activităţi - de pildă în comerţul obişnuit - am fi putut obţine profituri incomparabil mai mari. Dar nouă ne-a plăcut informatica şi CAD-ul!

R.: *D-le preşedinte, vă solicit un ultim comentariu: nu cred că acţiunile firmei A&C - sprijinirea învăţămîntului, ajutorul oferit beneficiarilor, satisfacerea unor nevoi stringente ale societăţii actuale din România - ar avea, în vreun fel, un caracter filantropic.*

C.D.: Observaţia este corectă. Nu este vorba de filantropie, ci de avantaj reciproc. Sîntem convinşi că anii ce vin vor confirma că modul nostru de lucru este cel corect şi optim.

Ne luăm aici la revedere de la amabilele noastre gazde.

Pentru look-ul occidental al firmei şi calitatea oamenilor din A&C, pentru aspiraţiile şi debutul lor de excepţie, ni se pare potrivită expresia pe care Horaţiu ne-a lăsat-o în Epistolele sale:

"Majores pennas nido" - au aripile mai mari decît cuibul.

Ne rămîne să le dorim un cuib pe măsura aripilor lor!

A consemnat Sorin Golopenţa

Advanced Computing Environment

Consorţiul ACE, o grupare de peste 20 de ofertanţi de hard şi soft, s-a înţeles în aprilie '91 asupra unor noi standarde în domeniul PC-urilor şi calculatoarelor RISC. Sînt sprijinite procesoarele x86 Intel şi procesoarele RISC produse de MIPS, precum şi o versiune unificată de Unix de la SCO şi OS/2 3.0 ca sisteme de operare.

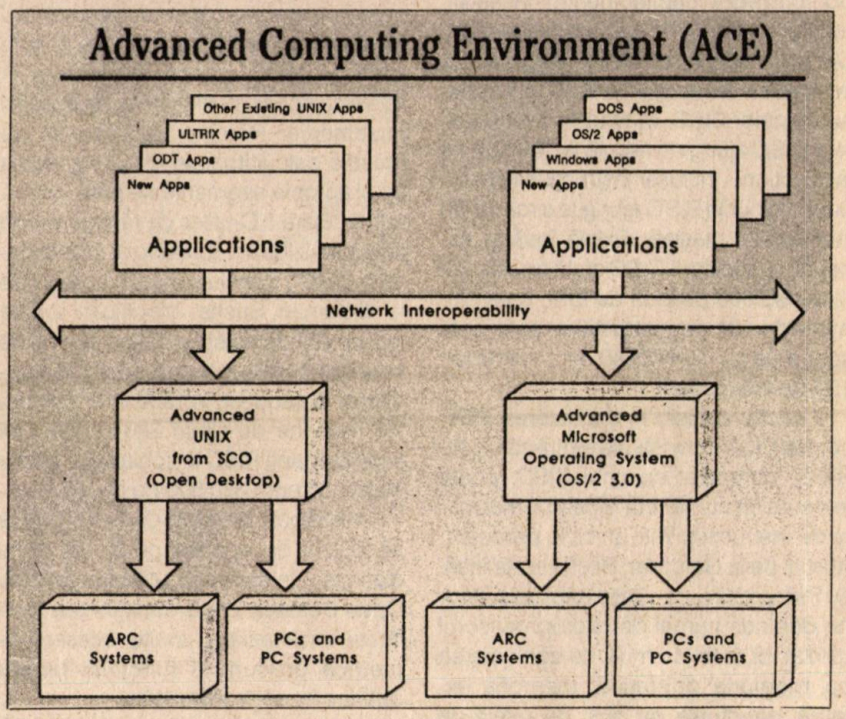
După cum se ştie, ramura calculatoarelor nu e săracă în foruri, consorţii şi comitete de tot felul. De la data de 9.04.1991 li s-a mai adăugat o nouă iniţiativă de ramură, care promite mult pentru viitor. În iniţiativa ACE (Advanced Computing Enviromment) s-au grupat 21 de întreprinderi de frunte ale industriei calculatoarelor, care împreună realizează un dever anual de 50 miliarde dolari.

ACE şi-a fixat ca obiectiv un mediu tehnologic de vîrf, bazat pe două platforme hard şi două medii - sistem de operare. Ambele sînt acceptate ca fiind egal îndreptăţite.

Platforma hard ACE o constituie PC-urile pe baza procesoarelor Intel x86, şi ca specificaţie standard suplimentară, calculatoare bazate pe procesoarele RISC produse de MIPS. Sistemele de operare sînt un UNIX evoluat, Open Desktop, produs de SCO şi OS/2 3.0 produs de Microsoft.

Se anvizajează şi înglobarea bazei de reţele deja instalate la clienţi. Consorţiul ACE vrea să sprijine servicii de reţea, atît pe PC-uri cît şi bazate pe Unix. Lucru valabil şi pentru mediul SNA al IBM-ului şi DEC-net-ul de la Digital Equipment. Primele echipamente şi "development kits" pentru producători erau aşteptate pentru sfîrşitul anului 1991.

Utilizatorii finali vor putea probabil cumpăra încă din acest an primele produse finalizate, software şi staţii de lucru. Specificaţiile elaborate de grupul ACE sînt "deschise", şi orice producător de hard sau soft le poate obţine. Întreprinderile care doresc să dezvolte sisteme compatibile au acces neîngrădit la specificaţiile ARC (Advanced RISC Computing) şi pot participa la comitetele tehnice.



Declaraţia de intenţie a ACE relevă un obiectiv de bază - compatibilizarea la nivel de sursă a programelor disponibile sub DOS, Windows, OS/2 şi Unix cu cip-ul RISC produs de MIPS. Aplicaţia trebuie doar recompilată pe suportul hard RISC. Compatibilitatea la nivelul codului sursă va fi garantată de amplasări similare ale octeţilor în memoria principală şi pe memorii externe la cele două tipuri de cipuri, Intel şi MIPS. Această tehnică poartă numele de "Little Endian Ordering". Astfel, iniţiativa ACE integrează PC-urile bazate pe 386 şi 486 în mediul ARC şi oferă ca obiectiv următor interoperabilitatea şi coexistenţa de PC-uri Intel şi sisteme ARC într-o reţea. Dar întrebarea decisivă este: cînd să se folosească un PC Intel şi cînd un sistem RISC. Cu aceasta s-ar ajunge din nou la problema multdiscutată la sfîrşitul anilor '80, dacă procesoarele RISC le sînt superioare celor CISC.

Membrii iniţiativei ACE

- The Acer Group
- Compaq Computer Corporation
- Control Data Corporation
- Digital Equipment Corporation
- Kubota Computer
- Microsoft Corporation
- Mips Computer Systems
- NEC Corporation
- NKK Corporation
- Olivetti Systems and Networks
- Prime Computer
- Pyramid Technology Corporation
- The Santa Cruz Operation
- Siemens AG/Automation
- Siemens Nixdorf Informationssysteme AG
- Silicon Graphics Computer Systems
- Sony Corporation
- Sumimoto Electrics Industries
- Tandem Computer
- Wang Laboratories
- Zenith Data Systems

RISC contra CISC

Acum doi ani, a apărut un studiu efectuat de analiştii de la Goldman Sachs, o întrepridere americană ce disponibilizează capital pentru investiţii, dedicat "viitorului microprocesoarelor". Concret, studiul urmărea răspunsul la întrebarea dacă cipurile RISC ar fi superioare celor CISC. În timp ce procesoarele CISC ajung de regulă la 100-150 de instrucţiuni - inclusiv instrucţiunile compuse - cipurile RISC ajung la circa 70-80 instrucţiuni maşină. Există însă şi excepţii: susţinătorii CISC arată cu degetul cipul RS/600 produs de IBM, care dispune de 184 de instrucţiuni, spre deosebire de un Intel 80386 care are "doar" 150 de instrucţiuni.

Punctul decisiv în dezbaterea RISC contra CISC îl constituie faptul că un cip RISC, comparat cu unul CISC, poate executa un număr considerabil mai mare de instrucţiuni într-un ciclu procesor. Autorii de la Goldman Sachs arată însă, cu îndreptăţire, că viteza unui calculator nu depinde numai de microprocesorul utilizat, ci, la fel de mult, de conlucrarea cu memoria operativă, memoria externă, structurile de bus, de ratele de transfer a datelor, etc.

În timp ce în trecut ofertanţii de calculatoare RISC s-au concentrat preponderent asupra prelucrării datelor ştiinţifice şi tehnice, ieşiri grafice şi aplicaţii de construcţie (CAD/CAM), în prezent s-au adăugat şi aplicaţii în domeniul bazelor de date. Cipuri RISC Îşi găsesc tot mai des drum spre acest domeniu al prelucrării comerciale a datelor. Calculatoarele RISC, disponibile deocamdată numai ca modele constructive, probabil că se vor orienta cu precădere spre staţii de lucru grafice, fileservere în reţele cu soluţii multiuser sau/şi aplicaţii multimedia. Rămîne de văzut în ce măsură calculatoarele RISC vor putea să-şi apropie segmente de piaţă importante. "Piaţa PC-urilor va rămîne pe un timp previzibil un domeniu al procesoarelor CISC", concluzionează analiştii de la Goldman Sachs. Modificări rapide nici nu sînt de aşteptat, ţinînd seama de cantităţile livrate în trecut: conform aprecierilor Dataquest, în 1990 Intel a vîndut 7,5 milioane de cipuri 386 şi 486. Cei zece ofertanţi RISC în schimb au vîndut împreună cam 390.000 de bucăţi.

Unix Open Desktop de la SCO este un sistem de operare pe 32 biţi şi se distinge prin suprafaţa utilizator grafică. Open Desktop oferă funcţionare multiuser, multitasking, multiprocesare simetrică precum şi funcţiuni pentru operare în reţele distribuite.

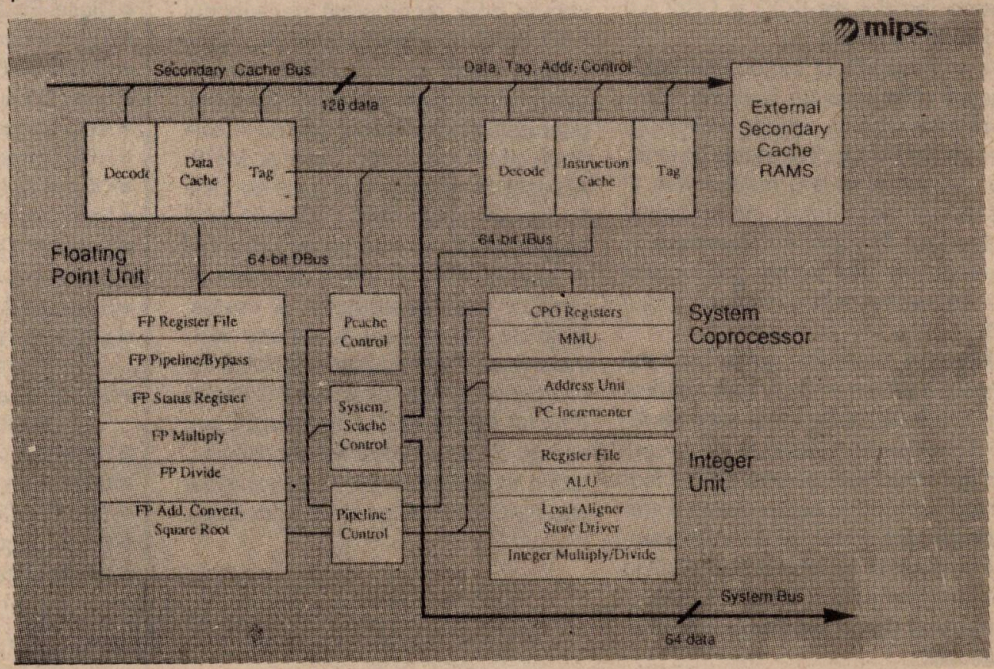
Un element de bază al iniţiativei ACE îl constituie specificaţiile ARC (Advanced RISC Computing). Ele urmează să garanteze compatibilitatea binară pentru softul aplicativ şi să formeze o platformă cît mai largă în sprijinul altor producători, care oferă echipamente periferice. Elementele de bază ale specificaţiilor ARC le formează procesorul RISC R4000 produs de Mips, dar şi tipurile R3000, R3000A şi R6000, interfeţe spre echipamente de I/O, funcţii ale subsistemelor de I/O precum şi formate de date pentru medii de memorare şi formate de fişiere.

Ambele sisteme de operare promovate de ACE sînt sisteme de operare pe 32 biţi. Ceea ce înseamnă că ele nu folosesc încă integral posibilităţile procesorului R4000, care lucrează pe 64 biţi. Aici mai rămîn probleme de rezolvat.

(I.F.)

Gang of nine

În 1988, nouă producători de calculatoare au definit standardul de bus EISA, concurînd "microcanalul" IBM-ului. Cei nouă au îndrăznit atunci, pentru prima dată, să introducă un standard pentru calculatoare MS-DOS care nu provenea de la IBM. Această faptă "rebelă" i-a adus consorţiului EISA porecla de "gang of nine" - "banda celor nouă". Unii din fondatorii EISA, precum ACER, Compaq şi Zenith, sînt şi membrii fondatori ai grupului ACE.



Cipul RISC R4000 este după datele furnizate de Mips primul microprocesor adevărat pe 64 de biţi. Construcţia schematică permite aruncarea unei priviri asupra modului său de funcţionare.

THE choice in software

AUTODESK
Microsoft
NOVELL
Lotus
Fox
BORLAND
Aldus
Corel
Symantec

雅仕 A&C INTERNATIONAL S.A.

Tel 40-0-53.53.15.

Standardul ACE devine realitate

Arată ca şi "Windows 3.1", are o arhitectură pe 32 de biţi, multiprocesor, şi funcţionează nu numai pe procesoarele Intel ci şi pe MIPS: este vorba despre sistemul de operare ACE "Windows NT" al firmei Microsoft.

Doar puţini au crezut că poate fi posibil: Microsoft a expus la Comdex '91 de la Las Vegas (Nevada) nu doar "Windows 3.1", ci şi "Windows NT", care fusese anunţat pentru mijlocul - sfîrşitul anului 1992. (NT = New Tehnology). Nu este vorba despre un urmaş al versiunii Windows 3.1, ci de un sistem de operare independent de DOS şi de sine stătător, care să coexiste alături de combinaţia "DOS/Windows 3.x", bineînţeles la un nivel mult mai înalt de posibilităţi şi de performanţe decît un DOS extins: o maşină cu un procesor i386 şi cu RAM de 8 MByte este configuraţia minimă absolută. În fig.1 se poate vedea care Windows pentru ce bază hardware este gîndit.

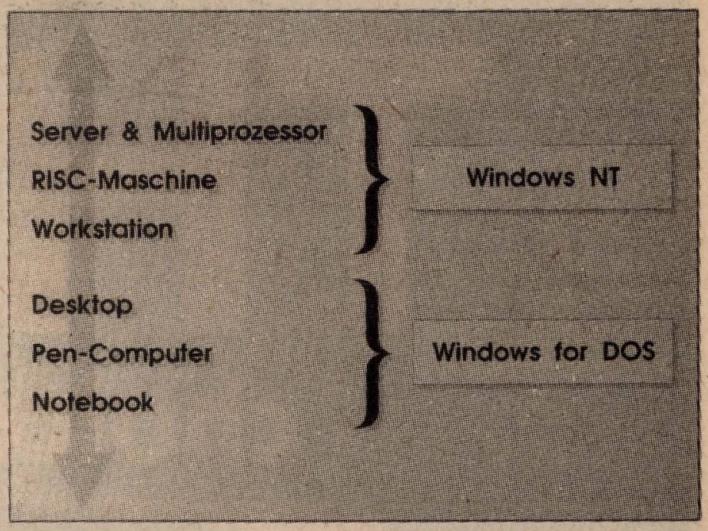


Fig. 1 Din grafic reiese clar faptul că Windows NT este gîndit pentru domenii de vîrf

Windows NT şi iniţiativa ACE

La Windows NT se lucrează încă din anul 1988, dar scopul de atunci a fost de a crea o nouă generaţie OS/2. Însă surprinzătorul boom al lui Windows a dus la o strategie pe termen scurt: în locul anunţării realizării sistemului de operare standard definit, în aprilie 1991, în cadrul cooperării ACE (Advanced Computing Environment), "OS/2 3.0", produsul a fost redenumit "Windows NT" iar "Presentation Manager" a fost înlocuit cu o suprafaţă Windows 3.1.

Însă structurile sistemului nu au suferit nici o modificare: el se bazează pe o arhitectură scalabilă de 32 biţi şi oferă un multitasking "pre-emtiv". În momentul de faţă sistemul funcţionează atît pe i386, i486 cît şi pe procesoare MIPS-RISC, susţine sisteme simetrice multiprocesor şi poate executa pe lîngă aplicaţii DOS şi Windows şi aplicaţii OS/2 şi Posix. Pentru un utilizator, acest lucru înseamnă că poate avea acces deplin la toată paleta de soft a 3 sisteme de operare incompatibile pînă în prezent - indiferent că lucrează pe un calculator Intel sau MIPS. Performanţele calculatorului vor fi exploatate optimal cu ajutorul arhitecturii scalabile şi a susţinerii sistemului multiprocesor.

Miezul sistemului

Structurile interne ale S.O. Windows NT (fig. 2) sînt la fel de complexe ca şi performanţele sale. La prima vedere se pot distinge 4 componente ale sistemului, dintre care 2 lucrează în modul de lucru sistem "privilegiat" iar celelalte două în cel "neprivilegiat".

Nucleul sistemului Windows NT, şi deci clasificat ca şi "privilegiat" este aşa-numitul "Executive", care este răspunzător de toate caracteristicile de bază ale sistemului cum ar fi managementul de memorie, multitasking-ul şi multiprocesarea, portabilitatea şi diferite mecanisme de siguranţă. El conţine şi aşa-numitul "Microkernel", care pune la dispoziţie toate funţiunile dependente de maşină într-un cod de doar 50 KByte. Acest Kernel este singura parte a sistemului de operare Windows NT care trebuie rescrisă la o transpunere a sa pe o altă platformă hardware. Interfaţa dintre Executive, şi toate celelalte componente ale sistemului se numeşte "Executive Service API".

(continuare în pag. 18)

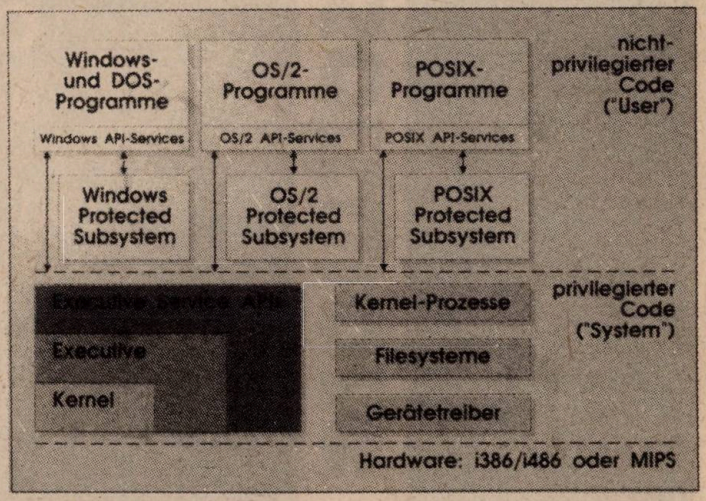


Fig. 2 Structurile clare ale lui Windows NT garantează securitatea, extensibilitatea şi portabilitatea sistemului

FOURTH SHIFT - sistem de conducere

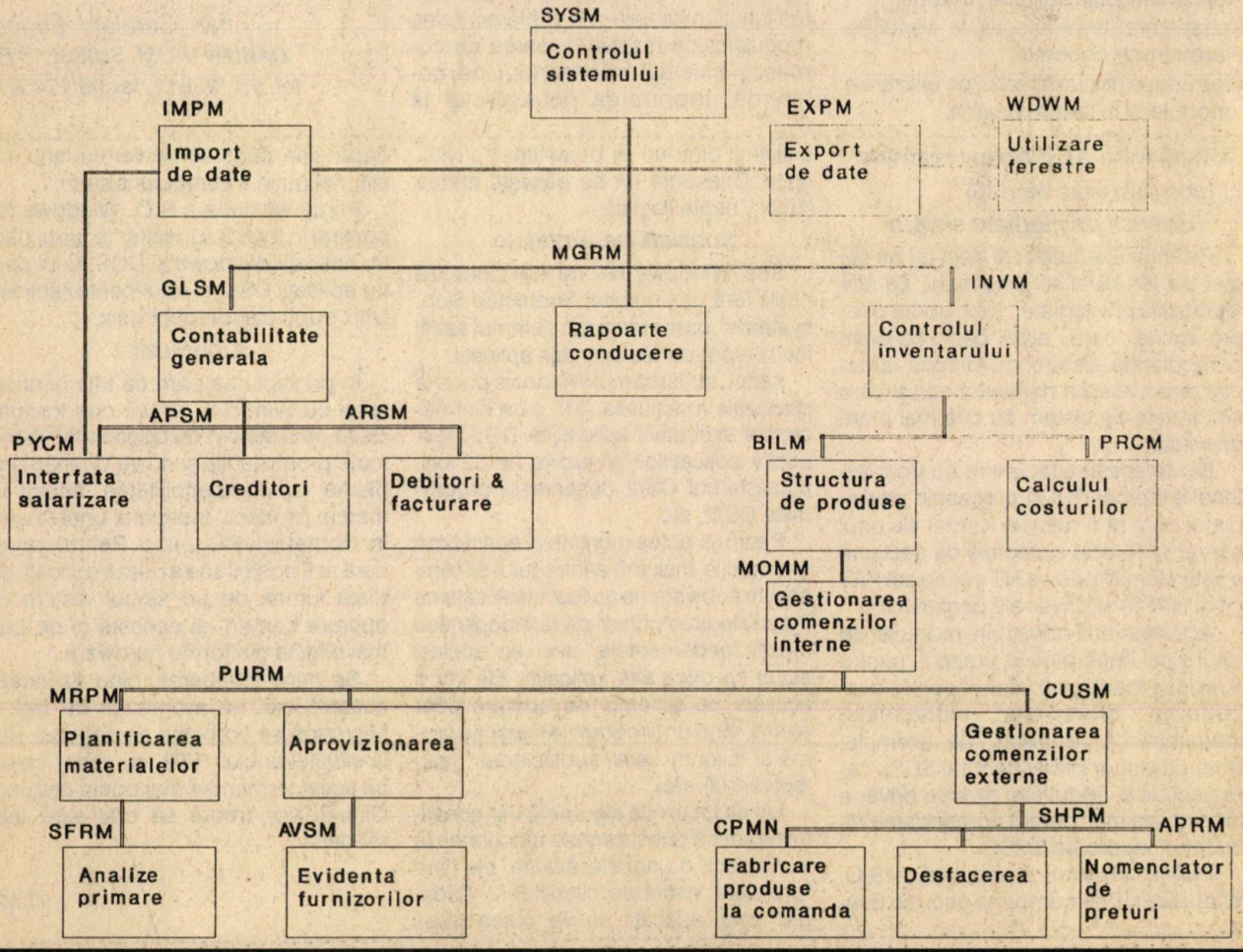
FOURTH SHIFT este un sistem care face posibile conducerea şi controlul mai eficient al afacerilor unei întreprinderi industriale. Este un sistem complet care nu se referă numai la controlul producţiei şi al inventarelor. Prin sistemul **FOURTH SHIFT**, problemele relative la activităţile de producţie şi cele financiare sînt legate între ele folosind aceleaşi tranzacţii şi date cu care întreprinderea operează în cadrul realizării planului. **FOURTH SHIFT** face posibil ca obiectivele de conducere şi de producţie să fie traduse în informaţii pe care fiecare persoană din întreprindere să le poată înţelege şi folosi.

FOURTH SHIFT este un sistem de conducere a unei întreprinderi în timp real. De provenienţă SUA, elaborat de către **FOURTH SHIFT Corporation** din Minneapolis, cuprinde soluţii de rezolvare performante, pe principii occidentale, pornind de la fundamente economice care stau la baza celor mai avansate teorii de specialitate. Datorită flexibilităţii şi adaptabilităţii, sistemul include soluţii generale aplicabile departajat situaţiilor concrete ce pot apare în pregătirea şi realizarea procesului decizional şi de conducere la diferite nivele în cadrul unei unităţi economice.

Sistemul **FOURTH SHIFT**, acoperind o arie mare de activităţi, este conceput modular. Modulele componente ale sistemului înglobează funcţiuni grupate logic. Datorită faptului că sistemul trebuie abordat eşalonat în implementare şi exploatare, funcţiunile modulelor cuprind posibilităţi de extindere a ariei de realizare a unor facilităţi în cadrul unui modul. Astfel că înaintînd în implementarea modulelor, asemenea sarcini sînt preluate implicit de modulele aferente. Spre exemplu, o evidenţă completă cantitativă şi valorică a gestiunilor materiale se obţine în urma implementării modulului CONTROLUL INVENTARULUI, reprezentat pe nivelul II în diagrama următoare. Înaintînd în implementarea sistemului, unele funcţiuni privind evidenţierea ieşirilor de materiale din gestiuni sînt apoi preluate de modulul GESTIONAREA COMENZILOR INTERNE.

Pentru a face din **FOURTH SHIFT** un sistem accesibil, uşor de înţeles şi învăţat, fiecare modul în parte este furnizat pe dischete de generare însoţite de documentaţia aferentă: manual de utilizare, ghid de exerciţii, caseta video de instruire, baza de date de instruire şi broşura program suport pentru client.

Sistemul **FOURTH SHIFT** crează şi gestionează o bază de date proprie. Are de asemenea incluse soluţii proprii de securitate a datelor prin funcţiuni de acordare de drepturi în consultarea şi întreţinerea datelor pe baza unor criterii de identificare a utilizatorului şi a drep-



turilor acestuia pentru fiecare caz în parte. De asemenea, include procedee proprii de analiză a gradului de ocupare a spaţiului alocat bazei de date, adaptarea dinamică a spaţiului la volumul datelor, urmărirea erorilor şi rezolvarea incidentelor, fişiere de istoric şi arhivă a operaţiilor efectuate în sistem, etc.

Sistemul **FOURTH SHIFT** este realizat pentru lucrul în reţea de calculatoare compatibile IBM PC. Pentru a îndeplini toate condiţiile de securitate a datelor şi valorificare a posibilităţilor, sistemul necesită un server pentru reţea şi un server pentru baza de date, urmînd să se stabilească caracteristicile acestora în funcţie de condiţiile concrete ale utilizatorului. Rămîne la latitudinea utilizatorului, de asemenea, stabilirea numărului de staţii de lucru şi amplasarea acestora.

Faţă de forma iniţială a sistemului **FOURTH SHIFT**, firma **DARIAN Rom Suisse S.R.L.** din Cluj a realizat:

- adaptarea interfeţei cu utilizatorul prin prezentarea ecranelor de culegere şi consultare date în limba română,
- prezentarea mesajelor de Help şi control al execuţiei în limba română,
- adaptarea terminologiei la condiţiile economice concrete,
- furnizarea documentaţiei de utilizare a modulelor în limba română.

Prezentarea de pe pagina anterioară cuprinde evidenţiate o parte dintre modulele sistemului **FOURTH SHIFT**, pentru a sugera faptul că între module există o legătură logică şi o ierarhie în abordare:

În concluzie, sistemul **FOURTH SHIFT** permite abordarea eşalonată a funcţiunilor în cadrul diferitelor module. Utilizatorul poate opta pentru resortul în care doreşte să obţină rezultate de primă urgenţă. În funcţie de opţiunea privind varianta de abordare a modulelor, sistemul oferă garanţia că structura şi conţinutul datelor cerute pînă la un anumit nivel în implementare şi exploatare sînt acoperitoare pentru rezolvarea oricărei alte funcţiuni a sistemului, cu singura restricţie privind ordinea logică de abordare conform schemei anterioare. Cu alte cuvinte, dacă la o unitate economică se doreşte, spre exemplu, rezolvarea în primă instanţă prin implementarea sistemului **FOURTH SHIFT** a unor probleme legate de evidenţa gestiunilor şi calculul costurilor, există garanţia că structura şi conţinutul datelor cerute şi utilizate în acest scop sînt compatibile şi folosite de sistem în extinderea funcţiunilor sale prin implementarea modulelor ce rezolvă lansarea de comenzi, calculul necesarului pe comandă, raportarea necesarului la stocurile existente, urmărirea realizării comenzilor şi încadrarea în consumuri. De asemenea, sistemul asigură introducerea informaţiilor din locul unde acestea sînt generate. Se depăşeşte astfel inconvenientul centralizării în anumite puncte de prelucrare a documentelor suport de informaţii.

Este de menţionat faptul că sistemul **FOURTH SHIFT** reprezintă un instrument de bază pentru conducerea unei întreprinderi economice. În utilizarea acestui instrument însă, utilizatorul se confruntă cu probleme inerente legate de organizarea internă a unor activităţi în cadrul întreprinderii, cerinţe de adaptări de fluxuri informaţionale, creări şi urmăriri de responsabilităţi în întreţinerea şi exploatarea datelor, decizii organizatorice corespunzătoare, etc. Este bine să se aibă de la început în vedere că sistemul **FOURTH SHIFT** este un instrument deosebit de complex care trebuie exploatat cu pricepere şi profesionalism şi ale cărui rezultate depind direct de oportunitatea şi corectitudinea deciziilor luate în implementare şi exploatare.

mat. Caraiani Eugenia
DARIAN ROM SUISSE S.R.L.
tel. 95/123611, fax:95/124567

Standardul ACE devine realitate

(continuare din pag. 16)

Extensii privilegiate sistem

Pe lîngă Executive, şi deci tot pe cel mai de jos nivel al sistemului, se află "extensiile privilegiate". Este vorba despre rutine, care, altfel decît celelalte componente sistem, au dreptul la acces direct asupra hardware-ului şi care sînt tratate de sistem cu cea mai mare prioritate.

Dintre acestea fac parte nu doar extensiile Kernel-ului şi procesele privilegiate, cum ar fi rutinele Kernel ale unui server LAN, ci şi sistemele de gestiune a fişierelor (Windows NT deţine atît FAT cît şi HPFS) şi driver-ele perifericelor.

Acestea vor fi construite modular, cu totul altfel decît pînă în prezent, pentru a reduce foarte mult efortul pentru dezvoltarea driver-elor individuale. Înseamnă că va exista, de exemplu, doar un singur driver general SCSI, care accede la rîndul său diverse driver-e individuale mici pentru comandarea diferitelor adaptoare SCSI.

Partea "Sistem", privilegiată, a S.O. Windows NT se compune deci din Executive şi dintr-un şir de extensii privilegiate. Deasupra lor se găseşte partea "User", neprivilegiată.

Subsisteme protejate

Sub Windows NT nu funcţionează nimic fără aşa-numitul "Protected Subsystems", care aranjează sistemul astfel încît se pot derula diferitele aplicaţii.

Astfel, subsistemul Windows pune la dispoziţie funcţiunile API, care sînt necesare executării aplicaţiilor DOS, respectiv aplicaţiilor Windows pe 32 biţi, subsistemul OS/2 deserveşte programele OS/2, etc.

Pentru a putea menţine în acest timp o protecţie maximă a întregului sistem, aceste subsisteme au fost clasificate ca "neprivilegitae", chiar dacă îndeplinesc funcţii fundamentale, deci au acelaşi statut ca orice altă aplicaţie. Ele vor fi activate de sistemul de operare doar atunci, cînd un program în execuţie (numit şi "client") cere "subsistemul" (sau "server"-ul) său.

Un alt factor de siguranţă l-ar constitui faptul că subsistemele funcţionează în domenii de adrese diferite, ele fiind "protejate" astfel de clienţii lor. "Căderea" unei aplicaţii nu va putea trage după sine căderea serverului său, cu atît mai puţin a întregului sistem.

Prima versiune a S.O. Windows NT conţine în total 3 subsisteme: unul pentru aplicaţii Windows şi DOS, unul pentru aplicaţii OS/2 şi unul pentru aplicaţii Unix după standardul Posix.

Concluzii

În principiu se pare că Microsoft va reuşi cu Windows NT să dea lovitura. Dacă noul sistem de operare îşi va ţine toate promisiunile şi nu va da mari probleme cu compatibilitatea, atunci va marca, probabil, începutul unei noi ere în domeniul PC-urilor. Pentru prima oară ar fi posibil să se ruleze aplicaţii din toată lumea pe un singur sistem de operare comun, şi aceasta şi pe cele mai diferite platforme hardware.

Se pune întrebarea, cînd va începe această eră? În momentul de faţă la Microsoft se vorbeşte de mijlocul pînă la sfîrşitul anului 1992, cu specificarea, că acest termen se mai poate prelungi. Clienţii vor trebui să mai aibă încă răbdare.

(I.M.)

A Bold New Shape. A Fast New Pace.

...re smooth. Our ...l thanks to ...y and ...d

Take a bold first step to reshape plotting at your firm. Find out more today: Call **800-932-1212.** In Canada, call 416-635-9010. We're expecting to hear from you.

雅仕

A & C INTERNATIONAL S.A.

AUTHORIZED DEALER

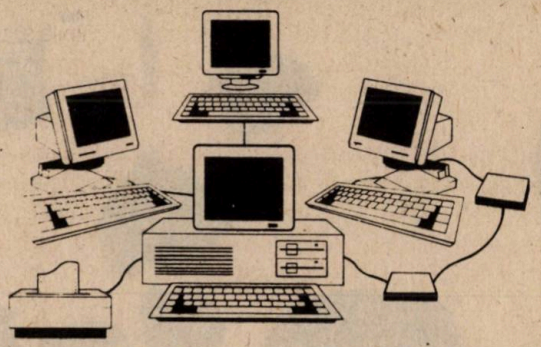
Constantin Tanase Str., No. 15, Sect. 2, Bucharest, 73299 Romania

Tel: 40-0-53.53.15
Service - "Hot Line": 40-0-12.77.74
Fax: 40-0-12.77.73
Telex: 10674 MAXIM R

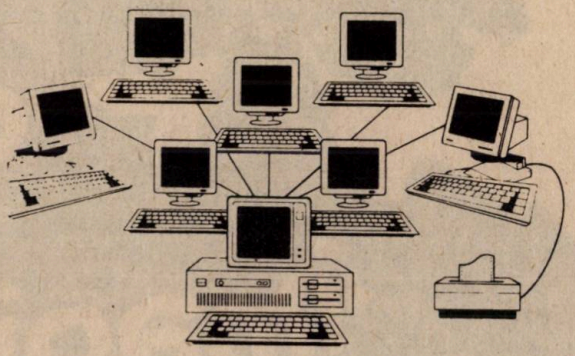
CalComp
A Lockheed Company

PC-MOS platformă multiuser DOS

Creşterea puterii microprocesoarelor a determinat folosirea sub capacitate a unui PC cu un singur utilizator. O cerinţă precisă într-o aplicaţie concretă constă în a determina numărul de posturi de lucru care cooperează pentru gestiunea informaţiei. Decizia de arhitectură posibilă trebuie să ţină cont de 3 factori: cost/punct lucru, performanţe, întreţinere sistem. Aceste considerente conduc în multe cazuri la opţiuni de tip multiuser. În cele ce urmează vă prezentăm soluţiile firmei The Software Link (TLS) în domeniul multiuser, a reţelelor din clasa Zero Slot Lan şi a deschiderilor către Novell şi mainframe.



Five-user PC-MOS supports up to four nonintelligent workstations.



Nine-user PC-MOS supports up to eight nonintelligent workstations.

PC-MOS sistem de operare multiuser, multitasking

Soluţia arhitectuală se bazează pe sistemul de operare PC-MOS. Acesta este oferit în variante cu 1 utilizator, 5 utilizatori, 9 utilizatori, 25 utilizatori. Sistemul de operare se poate instala pe maşini de tip 386 SX, DX şi 486 SX, DX sau PS/2. Deşi există şi variante pentru 286 (cu un modul de management de memorie), costul ajunge la cel al lui 386 SX, nejustificînd alegerea decît în cazuri speciale. Generarea sistemului de operare se face deci pe o maşină de tip cel puţin 386 SX. Pot exista 2 opţinui de generare pe hard disk: numai sistemul PC-MOS în toate partiţiile specificate (maxim 16) sau un mediu dual DOS într-o partiţie (de exemplu C;) şi 3 partiţii PC-MOS. În acest caz, la boot-are, se alege modalitatea de lucru:

- boot-are DOS caz în care lucrăm cu C: de pe hard disk şi cu A: şi/sau B: de pe floppy disk.
- boot-are PC-MOS caz în care lucrăm cu C:,D:,E:,F: adică partiţia DOS C: şi cele 3 partiţii PC-MOS D:,E:,F: împreună cu A: şi/sau B:.

Să definim acum 2 aspecte referitoare la comenzile sistemului de operare şi la pachetele care rulează sub mediul PC-MOS. Din punctul de vedere al sintaxei comenzilor de operare, acest sistem seamănă foarte mult cu DOS. Adică comenzile DIR, TYPE, COPY, DEL şi multe altele au aceeaşi sintaxă ca în DOS. Deoarece există facilităţile de multitasking este normal ca să apară şi comenzi noi specifice lui MOS. Vă pun acum o întrebare: cîţi dintre voi în exploatare continuaţi să lucraţi în mod normal cu comenzile DOS? De obicei, cred că preferaţi lucrul cu un DOS shell de tip Norton Commander sau PC Tools sau Xtree Pro Gold, etc., după gust, acest lucru degrevîndu-vă de memorarea sintaxei unor comenzi la nivel de linie de comandă. Ei bine, şi în MOS lansînd shell-ul preferat, aţi rezolvat deja lucrul cu manevrele uzuale de fişiere, directoare, etc. menţinînd deci un mediu de lucru care vă este familiar. În ceea ce priveşte lucrul cu pachetele DOS să le împărţim pe acestea în 2 clase: cele care permit dezvoltarea de aplicaţii noi (instrumente software) şi cele care nu necesită decît exploatare (Shell-uri, procesoare de texte, pachete de editare publicistică, etc.). În ceea ce priveşte cea de-a doua categorie, lucrurile sînt clare, pachetele executîndu-se conform specificaţiei de utilizare. Instrumentele software de tip compilatoare, generatoare de aplicaţii de baze de date, ş.a. permit dezvoltări de aplicaţii noi (la cheie). În acest caz se desprinde o folosire dublă a sistemului multiuser: pentru dezvoltare şi pentru exploatarea aplicaţiei finale. Cum rezolvă MOS problema?

Sistemul de operare (kernel) lucrează în mod 386 protejat în timp ce task-urile sînt executate sub controlul PC MOS în modul virtual 8086, permiţînd execuţia din memoria extinsă. Gestiunea memoriei este efectuată de un driver special al sistemului de operare, specificat în fişierul CONFIG.SYS. Fiecare task dispune de o zonă de memorie de mărime variabilă, maximul alocat depinzînd de controller-ul video ales şi de alte periferice speciale ale PC-ului, putînd ajunge la zona afectivă utilizator de pînă la 640 Kb, deci mai mult ca în DOS-ul uzual. Cum arată o pornire a lui PC-MOS boot-at de pe hard disk?

După testele BIOS de pornire, şi alegerea ca boot a lui MOS, sistemul de operare va citi dătele specificate în fişierul CONFIG.SYS. Apoi va executa AUTOEXEC.BAT. Presupunînd de exemplu, că vrem să lansăm 3 task-uri, vom specifica 3 instrucţiuni de tip ADDTASK. În acest moment, se poate executa, opţional, un fişier .BAT de iniţializare asociat fiecărui task (deci pe post de AUTOEXEC.BAT în viziunea DOS). În cadrul task-ului, utilizatorul lansează pachetul de tip DOS dorit. Chiar dacă programul DOS nu este explicit proiectat pentru folosire multiuser, se poate executa aceeaşi copie în task-uri diferite. De exemplu, un Wordstar care a fost lansat de la 2 puncte de lucru, va permite celor 2 utilizatori să lucreze nestingherit cu singura restricţie să nu folosească acelaşi fişier text simultan. Dar cum ar trebui să arate o aplicaţie proiectată special pentru a exploata accesul la fişiere în exploatare multiuser? În acest caz aceasta nu este o problemă a lui PC-MOS, ci mai degrabă a dumneavoastră, care trebuie să prevedeţi operaţiile de Lock(Unlock) la nivel de fişier sau înregistrare. Proiectînd cu acest scop în minte, lansînd de la mai multe puncte de lucru aceeaşi aplicaţie, veţi putea acceda şi simultan fişierele bazei de date.

În ceea ce priveşte compatibilitatea cu pachetele DOS deja existente, un studiu al revistei PC WEEK noiembrie 1989 comparînd această caracteristică notează MOS cu cea mai bună notă respectiv 9.0 la acest capitol. Nu se poate da un răspuns cert la această întrebare, deoarece numărul de pachete este uriaş. Totuşi, dintre cele care nu merg în modul protejat, un exemplu este Windows 3.0. Ca o explicaţie sumară, acest mediu, a fost proiectat ca o relaţie de tip gazdă-client. Dar Windows 3.0 deşi gazdă, nu este el însuşi client, cu alte cuvinte el permite altora să ruleze sub el, dar nu şi el să fie rulat de alţii. Trageţi singuri concluziile. În cazul unor pachete de acest fel, se pot instala în partiţia DOS, şi executa ca atare, deci numai pe terminalul calculatorului gazdă lucrînd cu partiţia disc C:.

Pentru "liniştea" dumenavoastră pot spune că Paradox, Foxbase, Dbase, Clipper, Ventura, PageMaker se pot folosi sub PC-MOS. Să revenim la explicarea facilităţilor sistemului PC-MOS. În primul rînd, el este multitasking. Numărul de taskuri lansate este teoretic nelimitat, practic el fiind limitat de memoria disponibilă şi de degradarea performanţelor în momentul existenţei unui număr mare de task-uri. Sistemul de operare realizează time-slice-ul pentru fiecare task, acesta putînd fi modificat dinamic. Există de asemenea şi 8 nivele de prioritate care pot fi schimbate dinamic. Lansarea sau oprirea unui nou task se poate face în orice moment. În acest scop există un TSR special activat de combinaţia <CTRL>+<Space>. Mărimea unui task este de la 32 Kb pînă la maximum posibil alocat de PC-MOS după boot. Dintre task-uri distingem 2 clase: cele care rulează în faţă (adică efectul lor se poate urmări la un terminal) şi cele care rulează în spate. O facilitate interesantă este aceea că putem trece oricînd de la un task la altul (switch) apăsînd <Alt>+<număr task> acest lucru permiţînd, de exemplu, ca un utilizator să lanseze o aplicaţie de la terminalul său şi apoi să comute în alt task lucrînd cu o altă aplicaţie, sau, mai spectaculos, comutînd într-un task "faţă" al altui utilizator, să vadă pe ecranul lui ceea ce lucrează celălalt.

Acest lucru poate fi extrem de atractiv într-o aplicaţie de tip educaţional cînd elevii comută în task-ul profesorului urmărindu-l de la terminalele lor, dar, şi profesorul poate comuta pe rînd în task-ul fiecărui elev, urmărind lucrul individual.

În ce priveşte folosirea fişierelor de tip Batch acest lucru este esenţial avînd în vedere că se pot lansa diverse clase de aplicaţii. Pentru automatizarea lansărilor se pune la dispoziţie un set de instrucţiuni de tip batch care, în esenţă, include instrucţiunile din DOS dar are şi facilităţi în plus: aşteptare tastă într-un domeniu specificat, texte în diverse culori, execuţia unor comenzi în subclase de fişiere (Only, Except), etc. Lucrul cu comenzi MOS individuale la nivel de prompter are facilitatea de a putea apela comenzile anterioare fără a fi necesară reintroducerea. Cum rezolvă PC-MOS partajarea resurselor de tip imprimantă sau coprocesor? În primul caz toate comenzile de tipărire lansate de la mai multe task-uri către aceeaşi imprimantă(e) sînt tratate de un task special (spooler-ul) care le gestionează conform dispoziţiilor de prioritate, şi a claselor de tipărire specificate (adică caracterele de control de iniţializare şi terminare a tipăririi fişierului corespunzătoare imprimantei). Starea de tipărire a unui fişier se poate vedea printr-un acces la task-ul spooler, informaţia fiind afişată într-o fereastră avînd la dispoziţie şi comenzi manuale de oprire/repornire tipărire. În ceea ce priveşte coprocesorul printr-o simplă declaraţie, problema este rezolvată. Adăugarea unor periferice nestandard ca de exemplu un Fax board sau un streamer nu necesită decît programarea unor batch-uri speciale care să ofere acces exclusiv. Lucrul cu mouse este posibil în anumite configuraţii, prezentate ulterior.

O altă facilitate este cea referitoare la accesul confidenţial la directoare şi/sau fişiere. Această problemă este rezolvată la nivel de sistem de operare prin parole şi definirea drepturilor de acces ale fiecărui utilizator.

Pentru comunicare de date între 2 utilizatori lucrînd la terminale diferite se pune la dispoziţie o facilitate numită PIPE care constă în transmiterea de fişiere între aceştia, consultarea fiind efectuată la cerere.

O menţiune specială trebuie făcută în cazul în care doriţi să dezvoltaţi o aplicaţie sub PC-MOS exploatînd facilităţile multiuser şi multitasking. În acest caz, se pot pune la dispoziţie documentaţiile necesare interfeţelor software spre PC-MOS iar instrumentele de lucru (compilator, asamblor, biblioteci) pot fi alese dintre cele existente în lumea DOS.

Clase de arhitecturi multiuser

Acestea sînt determinate de calitatea interfeţelor seriale şi corespunzător de terminalele folosite. Să le luăm pe rînd. În ceea ce priveşte interfeţele seriale, se pot distinge mai multe tipuri:

- a) Interfeţe seriale asemănătoare cu COM1 şi COM2 de pe PC care au pe modul I/O numai circuitele seriale lucrînd în întreruperi. De obicei toate căile (uzual avem multiplicitate de

4,8,16) folosesc aceeaşi linie de întrerupere dintre cele disponibile pe PC. În acest caz, este clar că performanţele sînt reduse, din cauză că modulul consumă din timpul procesorului, gîtuirea putînd apare cînd se transferă pachete mari de date pe viteze mari (pînă la 38,400 kbps).

¤ b) Interfeţe care au procesor pe modul dar nu consumă nici întreruperi, nici DMA. Acestea sînt aşa-numitele module inteligente care ascultă magistrala şi preiau datele care îi sînt proprii.

Sînt întîlnite şi variante intermediare celor prezentate, variabilitatea depinzînd de fabricanţi. Interfeţele din categoriile a), b), c) sînt în general cuplate cu terminale alfanumerice. Există şi module care pot fi cuplate la terminale grafice, de exemplu Hercules monocrom sau VGA, ele intrînd într-o categorie specială.

Deşi mulţi fabricanţi declară modulele oferite ca inteligente, fiecare soluţie trebuie clasată într-o anumită "zonă de inteligenţă".

Ca observaţie, amintesc faptul că pe o cale serială se pot cupla la aceste module nu numai terminale ci şi alte periferice seriale de exemplu imprimante, modemuri, alte echipamente specializate, etc.

Referitor la terminale, şi ele se pot împărţi în mai multe clase:

° A) Terminale alfanumerice din categoria familiară celor care au utilizat VT 52. În această categorie intră terminale care emulează Televideo, ADDS Viwpoint, Qume, Hazeltine, etc. Caracteristic lor este faptul că lansînd, de exemplu, un Norton sau un WS chenarele apar pur şi simplu ca nişte caractere oarecare. Concluzie: astfel de terminale sînt bune dacă am proiectat o aplicaţie specială pentru ele la care am controlat explicit gestiunea ecranului, mişcarea cursorului şi preluarea tastelor.

° B) Terminalele din categoria PC Term. Faţă de cele din categoria A), ele au tastatura identică cu a PC-ului şi recunosc setul extins de caractere afişabile permiţînd deci lucrul cu pachete de dezvoltare şi exploatare aplicaţii în modul alfanumeric monocrom.

° C) Terminale de tip Hercules monocrom. Acestea pot executa şi pachete grafice în standardul Hercules (720 x 348). Pachetele grafice DOS au driver pentru acest standard.

° D) Terminale de tip VGA. Practic nu avem nici o restricţie.

° E) Ca o categorie hibridă amintim cazul în care un PC poate fi utilizat ca un terminal, în anumite condiţii.

După o prezentare generală, independentă de PC-MOS să vedem cum suportă acesta combinaţiile de mai sus. Se pot utiliza interfeţe din categoriile a), b), c), d) iar terminalele trebuie împerecheate corespunzător. Pentru fiecare tip de interfaţă există drivere software speciale ca şi pentru terminalele de tip A-D.

De dragul simplităţii, facem abstracţie de conectori. Trebuie însă menţionată distanţa maximă pînă la care se poate cupla un terminal. Ca o regulă generală, nu poate depăşi 150m. În cazul în care se doreşte cuplare la distanţă, acest lucru se face prin modem. Un PC se poate folosi de sine stătător, dar, printr-un pachet numit PC Emulink, se transformă în terminal al PC-ului central, cuplarea făcîndu-se la 38.400 kbps şi în modul CGA. În acest moment, PC-ul devine terminal avînd acces la resursele centrale. Se pot transmite fişiere la/de la PC-ul central rulînd MOS la cel local rulînd Emulink.

Soluţia de tip Hercules monocrom poartă numele de MaXtation SH-4. În acest caz, modulul de interfaţă are 4 căi şi este de tip inteligent. El este legat cu 4 cutii pe care se aşează terminalele Hercules monocrom. Fiecare cutie are cuplat pentru display, tastatura, mouse de tip IBM compatibil şi imprimanta paralelă, ceea ce conferă un mediu bun pentru pachete grafice. Se pot cupla maximum 4 module SH4 pe o gazdă, deci putem avea 1+16 puncte de lucru de acest tip.

Soluţia de tip VGA poartă numele de VGNA şi constă dintr-un modul care acceptă maxim 2 terminale VGA şi se conectează într-un mod special, putînd avea inclusiv mouse şi imprimanta locală.

Bineînţeles se pot face şi combinaţii: MaXtation SH4 cu terminale seriale şi PC Emulink sau VGNA cu terminale seriale, etc. În acest caz, fiecare punct de lucru are definit modul de lucru cu ecranul adică: mono, Hercules, CGA, VGA, etc.

Ca o concluzie la acest capitol (pe care nu am pretenţia că l-am epuizat), pot spune că alegerea corectă în funcţie de aplicaţia concretă, depinde de module, terminale (eventual PC-uri pe post de terminale) care dau un anumit joc de preţ/performanţe. Oricum, este clar că preţul pe punct de lucru este sub preţul unui LAN cauza principală fiind folosirea terminalelor în locul PC-urilor. În ceea ce priveşte performanţa, pot afirma că accesul direct la hard disk este un alt atu al acestei soluţii.

Interconectări de tip Zero Slot LAN

Am văzut în capitolul precedent că putem obţine arhitecturi multiuser. Să numim o astfel de structură un 'ciorchine'. Se pun următoarele întrebări:

—De ce este necesar să avem mai mulţi ciorchini?

—Cum putem cupla mai mulţi ciorchini între ei?

La prima întrebare răspunsul este dat de faptul că un ciorchine poate avea un număr limitat de posturi de lucru. Folosind card-uri inteligente, numărul de posturi poate ajunge între 9-17 şi, în cazuri speciale, în aplicaţii scrise sub controlul mediului de dezvoltare de aplicaţii, chiar şi la 25. Ca o regulă personală, consider că numărul de 9 posturi reprezintă o barieră dincolo de care funcţionarea este la un nivel mai redus de performanţă. Răspunsul exact este foarte complex, el depinzînd de următorii factori: frecvenţa motherboard-ului, viteza de acces la disc, viteza de transfer a discului, calitatea interfeţelor pentru terminale, viteza de transfer a caracterelor între terminale şi PC-ul central la un nivel acceptabil pentru aplicaţie, alte perferice ataşate la PC-ul central, etc. În concluzie, dacă un ciorchine nu satisface toate cerinţele aplicaţiei, este necesară proiectarea distribuită, în 2 sau mai mulţi ciorchini.

O soluţie pe care o oferă TSL pentru interconectarea a 2 sau mai mulţi cior-

chini este în clasa aşa numită Zero Slot Lan. În această viziune, cum sugerează şi numele, nu este necesar un modul special pentru interconectare (Zero Slot) acest lucru făcîndu-se pe linii seriale (COM1,COM2) sau pe linii paralele, de exemplu calea LPTi.

Reţeaua TSL poartă numele de LanLink pentru legătura serială şi LanLink 5X pentru legătura paralelă. Nivelul de performanţă este dat de 115.000 Kbps pentru legătura serială, respectiv 500.000 Kbps pentru legătura paralelă, de aici şi numele 5X). Aceasta este viteza fizică de transfer a mediului. În realitate, datorită overhead-ului sistemului de operare viteza de transfer efectivă este mult mai edusă. Într-un benchmark efectuat de PC Magazine (aprilie '90) legînd 2 PC-uri; unul de 8 Mhz şi altul de 12 Mhz de tip 286, transferînd de 15 ori un fişier de 1 Mb în blocuri de date de respectiv 512 bytes, 4Kb şi 16Kb testele au arătat următoarele rezultate: 22,5 Kbps pentru serial, respectiv 123,5 Kbps pentru paralel reprezentînd vitezele reale de transfer ale reţelei de tip LanLink rezultat general valabil ca ordin de mărime şi pentru alte soluţii de tip Slot Lan (cazul paralel a avut cel mai bun rezultat). Ce concluzie putem trage?

Acolo unde traficul de informaţie nu este intens, o astfel de soluţie este justificată prin preţul ei redus.

Interconectările se bazează pe definirile de tip server şi satelit. Un satelit, legat prin LanLink poate avea acces la partiţiile de pe hard disk-ul serverului şi la imprimantele acestuia. Presupunînd ca pe ambii ciorchini avem instalat PC-MOS, resursele de memorie consumate sînt de 32 Kb pentru fiecare satelit, respectiv server. Din punct de vedere arhitectural, sînt posibile interconectări de tipul un server şi maxim 25 sateliţi sau un satelit poate fi conectat la maximum 8 server-e. Ca o curiozitate, se poate configura un PC astfel ca să poată fi şi satelit şi server pentru al doilea, şi invers, pe o aceeaşi cale de legătură. În acest fel ambii au acces la maximum de resurse ale celuilalt. Distanţa dintre ciorchini determină şi performanţele. Ca exemplu, pe linie serială pînă la 20m se poate merge pe 115.200 kbps. Acolo unde este necesară conectarea pe distanţe mai mari de 300m, legătura se poate face prin modem cu viteze de transfer între 300 şi 9600 bauds.

În cazul în care prin conectarea a mai mulţi ciorchini nu ajung căile seriale sau paralele, sînt necesare module suplimentare. Modulele de tip serial au viteze de pînă la 115.200 kbps iar cele paralele de pînă la 500.000 kpbs şi sînt prevăzute cu 4 căi.

Există şi o soluţie de conectare a unui Laptop la un ciorchine PC-MOS, soluţie numită LanLink Laptop realizînd transfer de fişiere, lucrul ca terminal, folosirea imprimantei de către Laptop, etc.

Deşi soluţiile TSL oferite sînt de tipul interconectări multiple, le putem asocia ca performanţe reţelelor din clasa Zero Slot Lan (în accepţiunea primară acest lucru însemnînd numai 2 PC-uri conectate).

O dilemă care apare este cea legată de alegerea soluţiei între multiuser şi LanLink. În general, soluţia multiuser este ideală în subgrupe de lucru întîlnite în departamente specializate: financiar, contabilitate, în timp ce LanLink poate interveni în legarea lor prin subseturi transmise între ele. O altă regulă este aceea că programele care lucrează intens cu CPU de exemplu, de tip CAD/CAM lucrează mai bine în mediu distribuit, pe cînd cele legate de acces intensiv la disc lucrează mai bine în mod multiuser.

Alte soluţii

În cazul în care se doresc performanţe superioare, se oferă posibilităţi de interconectare noi.

— Legarea către o reţea Novell Netware se face prin adăugarea modulului ales: Arcnet, Ethernet, etc. introdus pe calculatorul ciorchine. Este apoi lansat un task special numit PC Gateway similar cu IPX/NET3 care permite ca pînă la maximum 16 puncte de lucru să fie considerate din punctul de vedere al File serverului rulînd Netware ca PC uri nod pe reţea. În această soluţie, terminale cuplate prin modem la ciorchine reprezintă o soluţie ieftină (în loc de PC-uri), fără o degradare a performanţelor deoarece prin modem se transmit numai caracterele de ecran şi tastele (schimb redus de informaţii), task-ul pentru terminal fiind rulat local, pe ciorchinele PC-MOS.

— Sistemul este deschis şi pentru legături cu mini şi mainframe, de exemplu legare cu AS 400, IBM 3270 sau IBM 5250.

Concluzii

Ce resurse umane implică exploatarea unui mediu de lucru PC-MOS? Odată instalat şi configurat sistemul, exploatarea sa se face sub controlul unui administrator care gestionează fişierele batch pentru a permite lucrul în diverse aplicaţii, parolele de acces, a îmbunătăţi (eventual) dinamic rularea anumitor task-uri utilizator, a face salvări şi restaurări a anumitor fişiere pe dischete/casete, etc. Avînd în vedere că, odată pus la punct mediul de lucru, sarcinile acestuia sînt temporare, opinez că încărcarea sa este medie-redusă într-un singur ciorchine. La mai mulţi ciorchini, încărcarea devine mai mare ducînd la o alocare permanentă a sa.

Soluţiile prezentate reprezintă o alternativă de alegere a unui sistem informatic. Ele pleacă de la o premiză esenţială: trebuie să ştim ce vrem de la el, pentru a-l configura corespunzător. Totodată, din expunerea prezentată cred că am răspuns la cele 3 întrebări puse în prefaţă referitoare la cost/punct lucru, performanţe, întreţinere.

Prin deschiderile create, sistemul se poate dezvolta incremental, în paşi, începînd de la un ciorchine, apoi încă unul conectat, etc. Ceea ce este esenţial este faptul că se oferă soluţii deschise, protejînd investiţia iniţială şi efortul de a reconstrui noi aplicaţii dacă în viitor am opta pentru un drum alternativ nou.

ing. Denislam Cetin
Multipoint S.R.L.
tel./fax (90) 353335

Desktop Publishing

Odată cu apariţia versiunilor Windows ale programelor Ventura Publisher şi Pagemaker, PC-urile concurează MAC-urile şi în domeniul Desktop Publishing. În cadrul testului aceste programe au fost puse faţă în faţă cu Quark XPress, cel mai performant program DTP de pe MAC.

Pagemaker 4.0

"Pagemaker", programul DTP al firmei scoţiene Aldus poate fi considerat unul dintre strămoşii programelor DTP. Versiunea actuală 4.0 este disponibilă atît pentru Macintosh cît şi pentru PC, sub Windows. Specialiştii DTP afirmă că ambele variante se remarcă printr-o utilizare extrem de uşoară.

Instalarea nu pune probleme, dar sistemul trebuie să îndeplinească cîteva cerinţe pentru ca programul să poată funcţiona: astfel procesorul ar trebui să fie cel puţin un 386, iar prezenţa suprafeţei utilizator Windows 3.0 este obligatorie. Aldus mai recomandă ca memoria minimă să fie de cel puţin 3 MB. Dacă însă se lucrează cu documente mari, cu multe caractere de literă şi cu multe grafice abia de pe la 8 MB de memorie munca îţi face plăcere.

Pagemaker nu lucrează, asemenea contracandidaţilor săi, orientat cadru, ci simulează o masă de montaj. Utilizatorul plasează toate elementele pe care le doreşte pe hîrtie, deasupra sau dedesubtul foii şi le potriveşte în pagină. În acest mod o pagină poate fi nu numai foarte repede realizată ci şi modificată.

Acest mod de lucru are însă şi un neajuns: o plasare exactă a elementelor individuale este greu de realizat, astfel pe cît de rapid poate fi tehnoredactată o pagină "în mare", pe atît de înceată şi de obositoare este ajustarea detaliilor.

Editorul de tabele, prezent în furnitura de livrare, nu face cinste produsului. Pe de o parte utilizează un format document propriu, pe de altă parte nu dispune de despărţirea în silabe şi nu permite nici o rotire a textului cu 90 de grade. Impresia este că acest editor a fost adăugat în grabă doar ca o reacţie la funcţiile oferite de programele concurente.

Editorul de texte înglobat, în schimb, este reuşit. El permite nu doar o editare rapidă ci face şi o corectură a textului (pentru limbile engleză şi germană), importul de texte din toate procesoarele de texte curente putîndu-se face fără probleme. O problemă poate însă să apară şi în acest caz: combinaţia coloană, paragraf, grafic poate provoca probleme de kerning (ajustarea automată a distanţelor dintre litere în cadrul unui cuvînt şi a distanţelor dintre cuvinte). Probleme de kerning trebuie rezolvate manual, dar pentru aceasta trebuie experienţă şi se pierde mult timp.

Facilităţile grafice, în schimb, sînt excelente. El recunoaşte o mulţime de formate grafice şi la importul unui grafic, ca şi al unui text de altfel, el recunoaşte formatul fişierului fără a mai fi necesară o specificare explicită a acestuia. Imaginile importate mai pot fi prelucrate în ceea ce priveşte luminozitatea şi contrastul. Cei care doresc să realizeze documente color au nevoie de produse suplimentare, care să facă separaţia de culori, în momentul listării la o imprimantă Postscript. În cazul culorilor pure însă este posibilă şi o listare separată a acestora din program.

Deoarece funcţiile de bază privind tratarea notelor de subsol şi antetelor de cap şi de subsol de pagină lipsesc, tehnoredactarea cărţilor este dificlă.

Cele două versiuni, pentru PC şi pentru MAC, au acelaşi mod de operare şi aceeaşi paletă de funcţii, astfel încît cine a lucrat odată cu Pagemaker pe PC o poate face şi pe Macintosh fără probleme.

Quark XPress

Quark XPress este singurul program din test care nu are şi o versiune pentru PC.

Testul a fost efectuat pe un Macintosh II cu hard disk de 200 MB şi 5 MB RAM. Instalarea programului, ca de altfel şi a celorlalte aplicaţii de pe Macintosh, nu pune probleme.

Programul lucrează orientat pe cadre şi linii. Acestea sînt elementele de bază cu care se construieşte scheletul paginii, în care se inserează apoi toate celelalte elemente. Toate obiectele odată plasate pe ecran pot fi mutate cu uşurinţă fiind posibilă şi o grupare a acestora. Astfel este posibilă şi o "paginare" rapidă.

Lucrul în detaliu se poate realiza prin specificarea coordonatelor. Poziţionările pot fi astfel extrem de precise. Obiectele pot fi rotite cu unghiul dorit. De asemenea chenarele textelor pot fi plasate exact şi pot fi rotite, putîndu-se stabili şi culoarea de fond şi haşura fondului.

Dimensiunea caracterelor de literă este scalabilă în paşi de 1/1000 de la 2 pînă la 720 de puncte tipografice. De asemenea haşurile pot fi luminate sau întunecate cu pasul de 1%.

Iniţialele sînt generate automat, putîndu-se stabili numărul de litere iniţiale şi dimensiunile lor.

Modulul de grafică permite trasarea de dreptunghiuri, dreptunghiuri cu colţurile rotunjite, elipse, cercuri şi chiar poligoane. Ultimele pot fi modificate prin ştergerea sau inserarea de puncte. Editarea tabelelor se pate face fără probleme. La prima vedere şi facilităţile de culoare par să fie realizate profesional. Quark XPress oferă o mulţime de funcţii pentru prelucrarea culorilor. Aproape tuturor elementelor, indiferent că este vorba despre fond sau despre scriere li se poate stabili o culoare pură sau obţinută prin amestec. Pentru combinarea culorilor sînt disponibile trei modele (RGB, CYMK şi HSB). Importul de grafice color şi de imagini din alte programe se poate realiza fără probleme.

Rezumînd, Quark XPress este unul din cele mai complete produse în domeniul DTP fiind superior atît lui Pagemaker cît şi lui Ventura Publisher.

Un lucru însă este clar; pentru a putea utiliza eficient bogata paletă de funcţii a acestui program, utilizatorul trebuie să acumuleze mai întîi experienţă. Pentru primăvara acestui an firma producătoare, Prisma, a anunţat şi o versiune pentru Windows.

Ventura Publisher 3.0

Versiunea Windows a acestui program a apărut în martie 1991. Versiunea GEM, anterioară, îşi crease un bun renume, astfel încît noua versiunea a fost aşteptată cu curiozitate. Încă de la înce-

put este clar că versiunea GEM este mai rapidă.

Instalarea nu pune probleme, doar că în loc de un singur director se creează două: "Ventura" şi "Typeset". Xerox recomandă lucrul pe o configuraţie minimă 386, 2 MB RAM, dar pentru cei care vor să lucreze comod ar fi de recomandat un 486/33 cu memoria peste 8 MB.

Dar chiar şi în această situaţie programul este lent. Cine crede că de vină ar fi doar "lentul" Windows se înşeală deoarece în condiţii similare Pagemaker este mai rapid.

Încă de la veriunile anterioare Ventura îşi crease un renume mai ales în tehnoredactarea cărţilor. Programul lucrează orientat cadru. Versiunile GEM şi Windows sînt foarte asemănătoare. Versiunea Windows nu utilizează toate facilităţile oferite de mediu. Astfel nu este posibil un schimb de date via clipboard deoarece Ventura utilizează un alt set de caractere ASCII decît celelalte programe de sub Windows. Formatele fişierelor text sau imagine care pot fi importante nu sînt suficiente. Astfel lipseşte, de exemplu, Word for Windows.

Lucrul cu texte voluminoase este mult uşurat. Ventura dispune de funcţii pentru crearea automată a cuprinsului sau a unui index de cuvinte cheie şi permite tratarea notelor de subsol.

Din punct de vedere tipografic Ventura se află, fără îndoială, în faţa lui Pagemaker. "Kerning"-ul şi spaţierea sînt realizate flexibil. Spaţierea paragrafelor, dimensiunea caracterelor şi forma lor, se poate realiza foarte simplu. Un lucru trebuie avut însă în vedere: modificările influenţează întregul document şi nu există o funcţie "Undo". Crearea de tabele sau scrierea de formule nu pun probleme. În domeniul PC, Ventura rămîne în continuare unul din cele mai bune produse.

De curînd a fost prezentată beta-versiunea Ventura Publisher 4.0. Ea lucrează cu modelele de culoare YMCK, RGB şi HLS dar şi cu paleta de culori Pantone.

Pentru separaţia de culori şi prelucrarea imaginilor vor fi disponibile următoarele pachete suplimentare: VP Scan, VP Separator, VP Photo Touch şi VP Color Pro.

Alte funcţii noi, în cadrul editării de texte, vor fi funcţia de "Caută/Înlocuieşte", cea de verificare a corectitudinii scrierii, ca şi funcţia "Undo" (refă). De asemenea va fi posibilă stabilirea unui contur cu ajutorul unei linii poligonale şi importul unor obiecte via clipboard.

Concluzii

Opţiunea pentru un program DTP sau altul trebuie să fie determinată şi de categoria de lucrări care urmează a fi executate.

Astfel:

» pentru lucrări expresive, cu multe imagini, vă recomandăm Quark XPress. Precizia cu care acesta poate plasa obiectele în pagină este extremă şi facilităţile grafice sînt de asemenea excelente.

» pentru tehnoredactarea unor cărţi, deci acolo unde volumul textului este mare, etc, cele mai bune soluţii le oferă Ventura Publisher.

» dacă, în schimb, precizia tipografică este mai puţin importantă şi aveţi nevoie urgentă de lucrare atunci Pagemaker poate fi soluţia cea mai bună.

(R. M.)

Nume	Aldus Pagemaker 4.0	Ventura Publisher 3.0	Quark XPress 3.0
Producător/Distributor (în Germania)	Aldus Software, Hamburg	Ventura Software, Krefeld	Prisma, Hamburg
Preţ (în DM, inc. MwSt.)	2793	3397	2793
Date tehnice			
Hardware recomandat	386	486	MAC Ci
Spaţiu necesar pe hard disk	cca. 3,5 MByte	cca. 3 MByte	cca. 2 MByte
Suprafaţa utilizator	Windows	Windows	MAC
Memorie minimă	1 MB	1 MB	necunoscut
Memoria recomandată	> 4 MB	> 4 MB	> 4 MB
Lucrează în reţea?	nu	da (taxă supl.)	da (modul supl.)
Suprafaţa utilizator			
Listă unelte	da	da	da
Listă unelte liber poziţionabilă	da	da	da
Funcţie Zoom	da (25-400% în 6 trepte)	da (3 trepte)	da (scara dorită)
Funcţie Undo	da	nu	da
Apel funcţii via tastatură	nu	da	da (liber configurabilă)
Help interactiv	da	da	da
Vizualizare format	nu	da	da
Mascare caractere de control	nu	da	da
Text şi tipografie			
Nr. max. documente deschise	1	1	7
Modele predefinite	nu	nu	254
Ştergere a mai multor pagini	nu	da	da
Editor de texte propriu	da	nu	da
Dimensiune fonturi	4 - 650 puncte	0,5 - 999 puncte	2 - 720 puncte

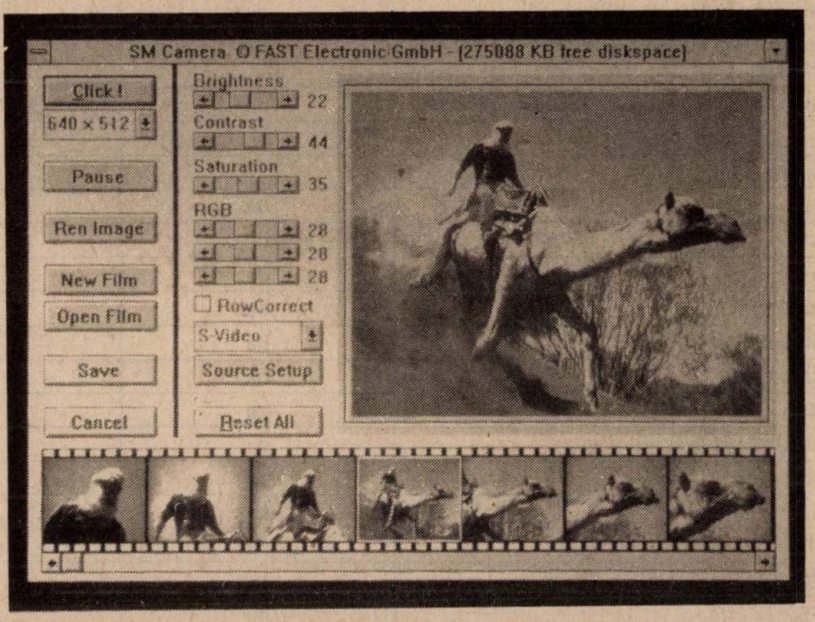
Nume	Aldus Pagemaker 4.0	Ventura Publisher 3.0	Quark XPress 3.0
Pas de modificare a dim. fonturilor	0,1 puncte	0,5 puncte	1/1000 puncte
Rotire text	paşi de 90 grade	paşi de 90 grade	paşi de 0,001%
Tabulatori	numerici / vizual	numerici	numerici / vizual
Iniţiale	da	nu	da
Verzale	da	da	da
Majuscule	da	da	da
Poziţionare exponent/indice	da	da	da
Algoritm automat de despărţire în silabe	da	da	da
Algoritm automat de despărţire în silabe pentru limbi străine	da	da	da
Kerning & Spacing	da	da	da
Editare tabele	da (funcţie grafică)	da	da
Nr. max. linii/coloane	80 linii, 80 coloane	9999 linii, 9999 coloane	da
Stabilire raster	da	da	da
Linii de înălţimi diferite	nu	da	da
Rotaţie text	nu	da	da
Definire format	nu	da	da
Editor de formule	nu	da	da
Definire blancuri	da	da	da
Tabele de kerning	da	da	da
Kerning manual	da	da	da
Funcţii layout			
Nr. max. pagini	999	9999	2000 pag. sau max. 2 GB
Nr. max. coloane	20	8	nelimitat
Coloane de lăţimi diferite	da	nu	da
Riglă	da	da	da
Poziţionare cu elemente de ajutor	da	nu	da
Poziţionare în coordonate XY	nu	da	da
Editor de formule	nu	da	da
Contur	da	între anumite limite	da
Antet şi subsol de pagină	nu	da	da
Tratare note de subsol	nu	da (automat)	da (manual)
Numerotare pagini	da	da	da
Tratare indexuri	nu	da	da
Realizare automată cuprins	da	da	da
Numerotare figuri	nu	da	da
Ataşarea unui text unei figuri	nu	da	da
Funcţii grafice			
Dreptunghiuri, Cercuri, Linii, Elipse	da	da	da
Funcţii vectoriale	da	nu	da
Tipuri de linii	11	10	3
Grosimea liniei selectabilă?	da în 8 trepte	da	da (selectabilă)
Diferiţi terminatori de linie	nu	da	da
Suprafeţe raster	da (7 diferite)	da	da (selectabilă)
Culori pure	da	da	da
Culori în amestec	da	nu	da
Modele de culori recunoscute	HLS, RGB, YMCK, Pantone	YMCK, RGB, HLS	YMCK, RGB, Pantone
Manipulări semitonale	da	da	da
Unghi haşură	da	da	da
Distanţă între haşuri	da	da	da

Nume	Aldus Pagemaker 4.0	Ventura Publisher 3.0	Quark XPress 3.0
Luminozitate	da	nu	da
Contrast	da	nu	da
Actualizare automată imagine	nu	da	începp. din vers. 3.1
Reprezentare cu nuanţe de gri la imagini half-tone	da	nu	da
Manipulare în dimensiune a graficii importate	da	da (specificare exactă)	da
Formate de import			
Grafică			
TIFF	da	da	da
PCX	da	da	da
WinDraw	da	nu	nu
Videoshow	da	da	nu
Tektronix Plot	da	nu	nu
ACAD ADI/PLT	da	da	nu
BMP	da	nu	nu
Lotus PIC	da	da	nu
Mac Paint	da	da	da
GEM	nu	da	nu
EPS	da	da	da
HPGL	nu	da	nu
RIFF	nu	nu	da
Mac PICT	nu	da	da
Texte			
ASCII	da	da	da
Word	da	da	da
RFT	da	nu	nu
Wordstar	da	nu	nu
Win Write	da	nu	nu
Word Perfect	da	da	nu
Wang	da	nu	nu
Samna	da	nu	nu
Symphony	da	nu	nu
PC Write	da	nu	nu
RTF	da	nu	nu
Excel	da	nu	nu
DCA	nu	da	nu
XY Write	nu	da	nu
Multimate	da	nu	nu
Lotus WKS/WK1, WK3	da	nu	nu
Formate pentru export			
ASCII	da	nu	nu
DCA/RFT	da	nu	nu
Word Perfect	da	nu	nu
RTF	da	nu	nu
Word	da	nu	nu
Opţiuni de ieşire			
Separaţie de culori	limitată	numai text şi grafica interna	da
Scalare la imprimare	da	da	da

Calculatoarele sînt capabile să vadă

Ca o continuare a articolului *"Bazele prelucrării de imagini pe calculatoare personale"*, din **IF 4/91(7)**, vă este prezentat în continuare sistemul de achiziţie şi prelucrare de imagini **Screen Machine** produs de firma germană **Fast Electronic.** Prototipul a fost prezentat publicului în cadrul prestigioasei expoziţii de tehnică de calcul de la Hannovra CeBIT martie 1990. Din octombrie 1990 şi pînă în octombrie 1991 firma vînduse deja 3.500 de astfel de sisteme devenind imediat leaderul european în domeniul respectiv.

Sistemul se compune dintr-o placă de achiziţii de imagini livrabilă pentru sisteme AT(ISA), PS/2 (Microchannel), MacIntosh Mac II şi din software de editare şi prelucrare imagini sub WINDOWS - 3.0 sau DOS (ca device driver). Imaginile digitizate pot fi afişate în culori reale (16.7 milioane de culori), chiar dacă placa grafică VGA dispune la un moment dat doar de 256 culori. Mai mult, se pot afişa în acelaşi timp imagini vii (live video), imagini statice (still video), imagini grafice (VGA) şi texte. Screen Machine depăşeşte astfel clasa sistemelor de achiziţie de imagine tip memorare cadru (Frame Grabber) fiind inclusă în sistemele tip Multimedia "Video in a Window", atît de utilizate la editarea video de publicaţii (Desktop Video). Poate că nu este de prisos o clarificare suplimentară a acestui concept nou şi pe plan mondial de "Multimedia". În primul rînd trebuie subliniat că acest concept nu reprezintă o nouă aplicaţie ci, o nouă tehnologie. Procesarea de texte şi de date cu calculatorul este un fenomen obişnuit, dar adevăratul potenţial vizual uman nu a fost încă pe deplin realizat.De îndată ce calculatoarele vor putea memora, reda şi manipula complet imagini video reale, toate aparatele casnice obişnuite: telefonul, televizorul, videorecorderul etc. vor putea fi înlocuite cu un calculator personal care, prin intermediul unei reţele de bandă largă va avea acces la o cantitate de informaţii aproape nelimitată. Cel mai important aspect referitor la acest nou mediu este generat de modul de lucru



Imaginile video pot fi preluate direct
şi pot fi convertite în oricare din formatele grafice uzuale

interactiv, ceea ce înseamnă că fiecare dintre noi va fi capabil, într-un mod sau altul, să transforme şi să influenţeze imaginile ce le vede pe ecran. Va fi posibilă selectarea, chestionarea şi deasemeni comunicarea cu acest mediu. Un astfel de mediu este excelent pentru transmiterea de cunoştiinţe şi pentru învăţare. S-ar putea ca nu peste mult timp această noua tehnologie să influenţeze viaţa noastră zilnică la fel cum telefonul sau televizorul au facut-o deja.

Din toate punctele de vedere mai sus expuse, **Screen Machine (SM)** este o placă Multimedia. Ea nu înlocuieşte cartela grafică VGA adăugînd la grafică şi text un nou mediu: imaginea video reală în orice poziţie şi mărime -comandată de calculator. Pentru a fixa exact locul acestei plăci în clasa sistemelor de achiziţie de imagini, trebuie precizat că tehnologia "Video in a Window" sau cunoscută şi sub numele de Overlay Board, este un pas intermediar spre sistemele pur digitale (tehnologia DVI C-Cube, MPEG (Motion Picture Coding Expert Group)-vezi **IF 2/91(5)**.

Caracteristici hardware

» Screen Machine dispune de trei intrări de semnal Video-Compus, astfel încît trei surse video avînd astfel de ieşire se pot cupla simultan chiar dacă au standarde diferite. Programul software permite comutarea automată între cele trei surse fără a fi nevoiţi să rearanjăm cablurile. Placa poate procesa semnale video în standardul PAL, NTSC şi opţional SECAM, recunoscînd automat felul semnalului video primit, adaptarea la acesta făcîndu-se automat. De asemeni SM suportă intrare S- Video (video profesională). Deoarece o intrare de acest tip foloseşte 2 din bornele de intrare (luminanţa + crominanţa), se poate conecta o sursă S-Video (profesională) cu o sursă de semnal Video Compus. Pentru a putea conecta şi surse RGB, firma **Fast Electronic** furnizează opţional un convertor RGB / S-Video prin care semnalul RGB este mai înainte transformat în semnal S-Video şi apoi digitizat.

» Tot ca intrare în SM este şi semnalul analogic de la controller-ul grafic VGA, legătura cu acesta făcîndu-se

KEYSYS

KEYSYS FAST Electronic Logitech Micrografx Microsoft

Your Distributor for:

LATEST VERSION

Become legal with Microsoft:

MS DOS 5.0 upgrade, installs on any previos MS DOS version include complete set of diskettes and manual in English

For only DM 180

LATEST VERSION

MS Windows 3.1

Includes complete set of diskettes and manual in English

For only DM 295

Fun with Micrografx:

Fun, Fast & Friendly, **Windows Draw Ver. 3.0**

Includes complete set of diskettes and manual in English

For only DM 255

Grab it with FAST Electronic:

ScreenMachine LITE, real time Video capturing board

Includes complete hardware and software under MS Windows

For only DM 1250

For your convenience, LOGITECH:

LogiMouse PILOT, 3 buttons, serial interface

Includes mouse and software

For only DM 87

KEYSYS Sisteme la cheie S.R.L.	**KEYSYS** GRAFX S.R.L.
Str. Arcu 13	Str. Călăraşilor 21/A
IASI, Tel: (981) 44499	ORADEA, Tel: (991) 32128

Prices are without ICM

Dealers welcome

printr-un cablu livrat împreună cu placa SM. Monitorul VGA la rîndul său se cuplează prin cablul existent la conectorul de ieşire analogică tip VGA existent pe SM. Această modalitate este necesară deoarece astfel SM se sincronizează cu semnalul video de la VGA, fiind capabilă să-şi afişeze imaginea în spaţiul grafic al unui cadru VGA. Mecanismul se numeşte **genlocking** şi este un sistem sofisticat compus dintr-o buclă PLL complet digitală şi un mixer video, iar rezultatul este remarcabil. Imagini venind de la videocamere, videorecordere, videodiscuri sau camere still video nu pun nici o problemă pentru Screen Machine, imaginea fiind de o stabilitate şi claritate deosebită. Şi din acest punct de vedere SM este superioară produselor similare aflate curent pe piaţă. Monitorul se conectează la o ieşire analogică distinctă a plăcii SM, fiind admise numai monitoare VGA (color cît şi monocrom) cu intrare analogică.

» Rezoluţia culorilor este de 21biţi/pixel pentru imagini live video şi 24biţi/pixel pentru imagini still video. Modelul intern utilizat este YUV, acest mod fiind foarte potrivit pentru prelucrarea de imagini.

Y este semnalul de luminanţă corespunzător luminozităţii (strălucirii) şi este digitizat cu rezoluţia maximă

U,V sînt componentele de culoare numite şi diferenţe de culoare, fiind digitizate la o rezoluţie mai joasă deoarece şi banda de frecvenţă în care sînt transmise în cadrul semnalului video este o bandă mai redusă decît cea a semnalului de luminanţă.

Formatul YUV este de două ori mai compact decît formatul RGB, necesitînd astfel numai jumătate din spaţiul de memorie cerut de un format RGB, chiar dacă semnalul RGB nu conţine informaţii suplimentare.

» Semnalul video este prelucrat în întregime în toată banda sa de frecvenţă de 5,5 MHz, frecvenţa de eşantionare fiind de 13,5 MHz. Această frecvenţă are avantajul prelucrării semnalelor cu standarde diferite fiind frecvenţa folosită de norma CCIR 601 (standardul videodigital).

» Sînt furnizate 3 rezoluţii de digitizare: 640x512, 320x256, 160x128. În general digitizarea unui cadru complet de la o sursă de imagine live este problematică datorită modului întreţesut de concepere al imaginii, obiectele în mişcare avînd poziţii diferite în cîmpul 2 faţă de cele din cîmpul 1. Rezultatul este un efect disturbativ numit 'lattice effect'. Screen Machine rezolvă această problemă printr-o comandă software de corecţie "Row Correct", cel de-al doilea semicadru(cîmp) fiind calculat prin interpolare din primul.

» Placa este dotată cu 1Mb de memorie RAM statică de 70ns (timp de acces) divizată în două zone egale: una pentru captura imaginii cealaltă pentru afişarea imaginii.

Optiuni hardware disponibile

1. **Interfaţa Audio (SM_Audio)**. Aceasta permite intrare sonoră de la toate cele trei surse video. Cele trei intrări stereo acceptă surse de semnal nu numai de la videocamere sau videorecordere, ci şi de la oricare altă sursă de semnal. Sînt prevăzute două ieşiri stereo capabile să comande tuner-e convenţionale sau amplificatoare; totodată semnalul poate fi amplificat la un nivel suficient pentru a acţiona difuzoare de tipul celor active, mici, dintr-o portavoce. Controlul modulului se face fie de sub Windows 3.0, fie din DOS. Sînt disponibile, de asemenea, seturile de subrutine ce comandă modulul Audio atît în C cît şi în Pascal.

2. **Optiunea Color Keying**. Constituie o altă cale de a reprezenta o imagine folosind asocierea biţi-pixeli (Bitmap). Cu această opţiune codul unui pixel nu reprezintă direct codul unei culori (culoarea digitizată), fiind mai degrabă un index într-o tabelă de culori "Color Look-Up Table" CLUT care memorează codul culorii actuale. Această tehnică, numită şi "color mapping " permite ca numărul de culori codificate de valoarea unui pixel să fie un subset al unei mulţimi mult mai mari de culori (paleta) reprezentată de biţi/pixel în CLUT. Acest subset de culori conţinut în CLUT poate fi schimbat în fiecare imagine afişată, sau, mai mult, chiar pentru fiecare linie în timp ce imaginea este afişată cum este cazul la SM. Astfel putem avea 8 bpp în bitmap şi 24 bpp CLUT, sistemul afişînd 256 de culori dintr-o paletă de 16777216 culori. Pentru a afişa imagini reale cu acest sistem de codare trebuie prelucrată imaginea cu un algoritm care să aleagă acele 256 culori care au cea mai mare rată de apariţie. Apoi fiecărui pixel îi este asignată o valoare index de intrare în CLUT pentru cea mai apropiată culoare de cea reală.

Atît codul culorii iniţiale asociate respectivului pixel cît şi valoarea culorii reale din CLUT sînt memorate la salvarea imaginii. La afişarea imaginii, valoarea culorilor reale se încarcă din CLUT în hardware, iar valorile iniţiale de pixel în memoria display (refresh). Cînd controller-ul de refresh citeşte un pixel pentru a-l afişa, valoarea acelui pixel este convertită în codul culorii reale prin interogarea CLUT-lui în timp real. Această tehnică permite totodată ca o culoare arbitrar selectată din CLUT să înlocuiască o anumită culoare dintr-o imagine live în toate poziţiile unde spotul întîlneşte această culoare. Se poate realiza astfel efectul de opacitate, se poate scrie şi "picta" într-o imagine live, de asemeni cadrul imaginii poate fi afişat pe ecran în forme arbitrare (de exemplu într-o sferă).

3. **Modul TV (Screen Machine TV)** care dă posibilitatea transformării PC-lui sau a MAC II-lui într-un televizor inteligent. Kit-ul Software care se livrează odată cu modulul SM-TV permite ajustarea automată a sunetului şi imaginii. La fel ca la reglajul de la distanţa se pot controla volumul, başii, balansul, contrastul, saturaţia, luminozitatea, fiecare din cele trei nuanţe primare de culoare toate prin intermediul calculatorului din Windows, DOS, sau prin program propiu. De asemeni se pot selecta şi memora pînă la 200 canale TV. Tot în acest modul a fost integrat şi un *decodor teletext*,care permite nu numai vizualizarea textului pe ecran, ci şi memorarea lui ca fişier ASCII sau PIC. Textul poate fi ulterior introdus într-un program de editare şi prelucrare de texte. Semnalul video de intrare (de la antenă) este recunoscut automat indiferent de normă (PAL, SECAM sau NTSC). Pentru a adăuga şi o bună audiţie în acest modul este incorporat şi un preamplificator cu trei intrări stereo şi două ieşiri stereo.

Software pentru Screen Machine

Programul central care coordonează funcţionarea plăcii se numeşte **SM-Camera** şi furnizează utilizatorului pe lîngă posibilităţile pe care un fotograf le are în camera de developare (darkroom) multe alte facilităţi. Se poate modifica luminozitatea, contrastul, saturaţia şi procentul culorilor primare. Sînt disponibile filtre pentru estomparea (softening) sau accentuarea (sharpening) contururilor. Se poate decupa numai o parte din imagine, mări şi micşora (zooming), retuşa sau decolora atît în întregime cît şi pe bucăţi. Imaginea poate fi rotită cu un unghi de 90, 180, 270 sau 360 grade.

Imaginea care rezultă în urma acestor transformări poate fi salvată pentru utilizări ulterioare în urmatoarele formate grafice standardizate:

DIB/BMP, Clipboard DIB, RLE, Clipboard RLE, RDIB, PMDIB/BMP, PCX, GIF, Aldus TIFF, TGA, JPEG 8-R8, Clipboard BMP, Aldus TIFF-LZW

şi în următoarele rezoluţii de culoare:

24 biţi/culoare (imagine reală)

8 biţi CLUT (256 culori)

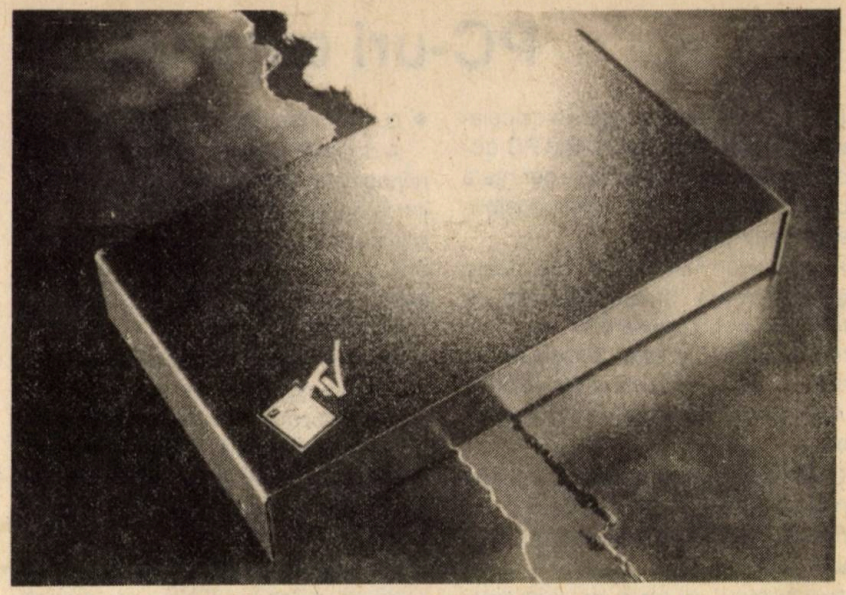
8 biţi gri (256 nuanţe de gri)

4 biţi-dither (16 culori)

1 bit dither b&w (imagini alb/negru)

Prin însăşi organizarea lor nu toate formatele grafice mai sus enumerate admit toate modurile de culoare amintite.

Imaginea poate fi comprimată cu un raport de compresie selectabil între 5:1 şi 50:1. Algoritmul de compresie este conform standardului JPEG (Joint Photographic Expert GROUP) de comprimare a imaginilor îngheţate (still), compresia imaginilor live conform standardului MPEG nefiind încă disponibilă. La un raport de compresie 10:1 nu sînt pierderi de calitate a imaginii vizibile. Algoritmul de compresie este scris în limbaj de asamblare 386. Versiunea pentru 286 este mai lenta. Deoarece o compresie a imaginii necesită o serie de calcule laborioase, timpul necesar compresiei depinde mult de viteza calculatorului gazdă. Pe un 386 la 25 MHz o compresie a unui cadru digitizat la rezoluţia maximă şi 24 bpp (480 Kb memorie) necesită 8s. Timpii de compresie şi decompresie sînt sensibil egali.



SM-TV permite controlul deplin al utilizatorului asupra oricărui tip de program de televiziune, direct din Windows 3.0, DOS sau de pe Macintosh

Nucleul de control SM-Camera poate fi executat atît de sub WINDOWS 3.0 cît şi de sub DOS (ca un device driver cu ajutorul unui program TSR-SMTSR). Astfel de sub prompterul DOS orice comandă spre SM are forma:

> *ECHO OFF comanda >SMDEV*

Este foarte simplu de scris fişiere de tip batch conţinînd şiruri de astfel de comenzi după un anumit algoritm. Sînt disponibile peste 125 de astfel de comenzi fiecare realizînd o anumită prelucrare sau interacţiune asupra imaginii.

Este disponibilă şi o bibliotecă conţinînd sub formă de apeluri aproximativ 125 funcţii de prelucrare, editare, încadrare, conversii, comprimări asupra imaginilor, inclusiv trasări de histograme. Biblioteca este scrisă atît în C cît şi în PASCAL fiind disponibilă şi pentru Windows DLL. Se poate comanda şi biblioteca de aplicaţii API (Application Programming Interface).

Produsele software pentru varianta MacIntosh conţin pe lîngă cele amintite mai sus o serie de interfeţe pentru:

SM-Desk Accesory, SM-XCMD pentru HyperCard, Supercard Plus, MacroMind Director, VideoBuilder şi Authorware, SM-X-Objects pentru MicroMind MediaMacker, SM-Plug-Ins pentru Photoshop, Quark-X-Press, PagTime, PageMaker, Color Studio, Image Studio şi Oasis, SM-Pocket Camera , Demo 4th Dimension External şi SM-Snap Shot XCMD.

Software-ul pentru Screen Machine IBM AT necesită 640 Kb RAM, DOS 3.3, sau versiuni superioare, Microsoft Windows 2.1, sau 3.0, mouse, monitor şi cartelă VGA.

SM Software pentru Mac II necesită cel puţin 4 Mb RAM, o cartelă grafică 8-sau 24-bit, monitor color sau monocrom. Software-ul este perfect compatibil cu 32-Bit Quick Draw.

Drosescu Radu
KEYSYS Sisteme la Cheie S.R.L.
Iaşi, tel. 981/46930,49064,
fax 981/44499
distribuitor unic în România
al firmei Fast Electronic

PC-uri extensibile (1)

Felul în care sînt concepute calculatoarele personale din seria IBM PC dovedeşte faptul că una din cerinţele impuse la proiectarea lor a fost posibilitatea de extindere.

Pînă la anumite limite, cineva poate adăuga memorie, drive-uri de disc, interfeţe sau coprocesoare în PC-ul pe care îl posedă. Limitele de extindere a diferitelor resurse pot constitui un criteriu suplimentar de alegere a unui calculator.

O altă posibilitate de extindere este prin înlocuirea unei resurse cu una superioară: mai mare, mai rapidă , mai funcţională.

Posibilităţile multiple de extindere, oferă posesorului unui PC senzaţia că îşi va putea rezolva, de-a lungul timpului, toate problemele care pot să apară, la preţuri mult inferioare preţului de achiziţie a unui nou PC. Această senzaţie este într-o anumită măsură justificată.

Lucrurile se complică atunci cînd chiar microprocesorul încorporat în PC stă "în calea fericirii" noastre. Microprocesorul încorporat defineşte, prin tip şi viteză, PC-ul. Spunem că avem un "AT 286 de 16 Mhz" şi apoi, dacă este cazul, dăm date despre memorie, disc, adaptor video, etc.

Tipul sau/şi viteza microprocesorului pot deveni un inconvenient în cazul în care o aplicaţie se execută nepermis de lent sau cînd se doreşte utilizarea unor programe care au fost scrise pentru un microprocesor mai puternic (de exemplu, programele scrise pentru 386 nu pot fi executate pe 286).

În prima situaţie, mai există şansa măririi vitezei de lucru prin metode hardware sau software: extinderea memoriei combinată cu utilizarea unor proceduri software de rapidizare a acceselor la disc (disc virtual, cache de disc), înlocuirea componentelor de memorie cu altele mai rapide pentru a elimina stările de WAIT, mutarea BIOS-urilor din ROM în RAM, înlocuirea discului cu unul mai rapid.

Cînd nu mai avem altă soluţie, sîntem puşi în faţa a două alternative:

- înlocuirea plăcii de bază
- cumpărarea unui nou PC.

A doua alternativă prezintă avantajul reînnoirii, la tehnologia momentului, a întregului sistem. Dacă, în plus, avem şi posibilitatea valorificării vechiului PC, prin vînzare sau utilizare în alt scop, această soluţie ar putea fi considerată cea optimă.

Schimbarea plăcii de bază are avantajul unui preţ mai redus, dar prezintă şi anumite dezavantaje: vechea placă este greu valorificabilă, după înlocuire se poate obţine un calculator mult mai prost cotat în clasa noului microprocesor.

Din cele prezentate pînă acum, se poate desprinde concluzia că, datorită faptului că permit adăugări sau/şi înlocuiri de subansamble sau componente în vederea creşterii performanţelor, PC-urile au ca trăsătură de bază extensibilitatea. De aceea, expresia "PC-uri extensibile" poate părea un pleonasm. Această expresie este justificată totuşi, în cazul PC-urilor care au fost proiectate plecînd de la ideea extensibilităţii procesorului, tocmai pentru a scoate în evidenţă această trăsătură remarcabilă. Ideea nu este nouă, dar a căpătat consistenţă în ultimii ani. Tot mai mulţi fabricanţi de PC-uri se orienteaza către realizarea unor variante extensibile. Unii dintre ei produc numai PC-uri extensibile. Acest proces este normal şi se datoreaza accelerării ritmului de apariţie a unor microprocesoare noi sau a unor versiuni mai rapide ale microprocesoarelor existente, ceea ce duce la scurtarea perioadei de uzură morală în condiţiile în care, perioadele de amortizare variază puţin sau deloc.

În general, PC-urile extensibile prezintă un raport preţ/performanţă comparabil cu cel al PC-urilor clasice (neextensibile ?), ceea ce face ca opţiunea cumpărării unui PC extensibil să fie atrăgătoare chiar şi pentru cei care nu au de gînd să-şi extindă vreodată calculatorul.

Dacă v-aţi hotărît pentru un PC extensibil ramîne problema criteriilor de decizie între tipurile de PC-uri extensibile existente pe piaţă.

Din punctul de vedere al preţului, trebuie luat în considerare, atît preţul configuraţiei iniţiale cît şi preţul extensiilor. Abordînd problema într-un mod grosier, putem spune că o extensie de la microprocesorul A, aflat în configuraţia iniţială, la microprocesorul B , trebuie să aibă preţul comparabil cu diferenţa dintre două PC-uri bazate pe cele două microprocesoare.

Mai pot interveni şi alte criterii (obiective sau subiective): cît se poate refolosi din componentele părţii care se înlocuieşte (prin vînzare sau mutare pe noul modul), evoluţia preţurilor PC-urilor sau/şi a modulelor de extensie.

Din punctul de vedere al performanţelor, trebuie avute în vedere, atît performanţele configuraţiei initiţle, cît şi cele ale configuraţiei obţinute prin extindere. Ideea este ca noul calculator, obţinut prin schimbarea procesorului, să fie bine cotat în clasa sa. Cu alte cuvinte, nu trebuie să comparăm noua configuraţie cu cea veche ci cu PC-urile din noua clasă în care a intrat calculatorul dupa extensie. Un exemplu negativ în acest sens este cel al extensiilor cu microprocesoare de 32 biţi care păstrează bus-ul de memorie de 16 biţi al configuraţiei iniţiale. În acest caz putem obţine, de exemplu, dintr-un 286 cotat onorabil între AT-urile 286, un 486 foarte prost cotat în clasa PC-urilor 486. Pentru a mai salva ce se mai poate salva, trebuie văzut dacă, rămînînd în clasa procesoarelor cu bus de 16 biţi (286, 3(4)86SX), nu se pot satisface cerinţele aplicaţiilor care au condus la necesitatea schimbării procesorului.

Un alt criteriu de decizie este cel al posibilităţilor de extensie pe care le oferă efectiv fabricantul (tipuri de microprocesoare şi viteze). Dacă fabricantul prezintă suficiente garanţii, puteţi lua în considerare şi promisiunile acestuia că în viitor va oferi module şi pentru alte tipuri de microprocesoare sau versiuni mai rapide , chiar şi pentru cele care sînt în faza de prototip la fabricantul de componente.

În unele cazuri, nu este de neglijat, ca şi criteriu de decizie, nici comoditatea instalării extensiei.

(continuare în pag. 55)

Fox Pro 2.0

DARIAN ROM SUISSE SRL
3400 CLUJ-NAPOCA
str. Observatorului nr. 146/15
tel. : 95-123611
fax: 95-124567

Pe o bază de date de 200MB, avînd milioane de articole, FOXPRO pe PC este mai rapid de 4 ori decît DB2/IBM pe un mainframe, iar diferenţa faţă de alte sisteme de gestiune ca Paradox sau DBase este enormă. Toate acestea datorită noii tehnologii de acces la date RUSHMORE.

Utilitarele sale de generaţia a patra, orientate pe obiecte, (generator de ecrane, de rapoarte, de meniuri şi proiecte) cresc extraordinar viteza de dezvoltare a aplicaţiilor. Şi toate acestea generează programe sursă FOX, accesibile utilizatorilor !

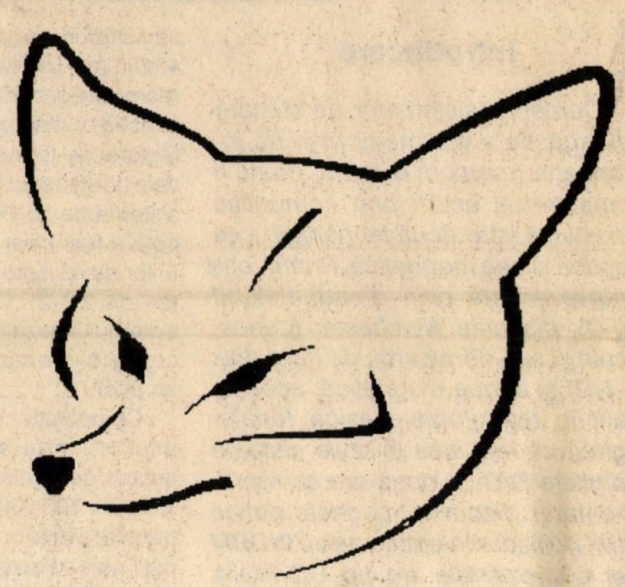
Ce este nou în FOXPRO 2.0 ?

Noul sistem de gestiune a memoriei asigură o exploatare optimă, astfel încît viteza creşte atît la sisteme mono- cît şi multiuser. Volumul indexării compuse reprezintă un sfert din cel al sistemelor similare şi totuşi este mult mai rapid.

RQBE / Relational Query-By-Example / asigură o interfaţă simplă cu ajutorul căreia, doar cu cîteva mişcări de mouse, obineţi informaţiile de care aveţi nevoie. FOXPRO va genera intrucţiunea SELECT/SQL corespunzătoare, care împreună cu celelalte comenzi SQL: INSERT şi CREATE TABLE pot fi folosite în aplicaţiile FOX.

FOXPRO 2.0 a fost dotat cu peste o sută de comenzi şi funcţii noi, ca de exemplu relaţia "one-to-many-to-many", noi metode de sortare, sau comanda BROWSE FOR care ar putea să fie ea singură baza unei noi aplicaţii.

Dezvoltarea proiectelor se realizează simplu şi eficient cu noul "project-manager", care urmăreşte fişierele folosite, înglobează în sistem toate modificările sau reconstruieşte întreaga aplicaţie. Generarea programelor EXE şi distribuirea produselor realizate este posibilă prin modulul opţional Distribution Kit.



Este posibil importul datelor dintr-o multitudine de formate : PARADOX, MULTIPLAN, LOTUS, SYMPHONY, EXCEL etc.

Prin produsele opţionale ale familiei FOX : Fox Graph, Distribution Kit sau Application Program Interface, Fox Pro 2.0 devine un instrument eficient şi complet pentru dezvoltarea aplicaţiilor.

Trimiteţi-mi dischetele demonstrative FOXPRO 2.0

NUME . . . . . .

FUNCŢIA . . . . . .

FIRMA . . . . . .

ADRESA . . . . . .

. . . . . .

TELEFON . . . . . .

FAX . . . . . .

Format dischetă 3,5" |_| 5,25" |_|

Dischetele demonstrative pot fi obţinute gratuit numai în schimbul acestui talon!

Medii optice şi magnetice pentru stocarea masivă a datelor.

O analiză comparativă a situaţiei actuale

Introducere

Domeniul sistemelor de stocare şi regăsire a volumelor mari de informaţie (memorii externe) poate fi caracterizat astăzi prin competiţia existentă între două tehnologii: cea optică şi cea magnetică. Prima este reprezentată prin discul flexibil (FD), discurile Winchester (cunoscute şi sub denumirea de hard disk - HD) şi banda magnetică, aceasta atît în tehnologia clasică (analogică) cît mai ales în noile sisteme digitale DAT. A doua are ca reprezentanţi discurile magneto-optice (MO), discurile optice de tip WORM şi dispozitivele de tip CD-ROM. Unele dintre aceste dispozitive permit atît citirea cît şi scrierea datelor (este cazul discurilor şi benzilor magnetice precum şi al discurilor magneto- optice) în timp ce altele permit, de regulă, doar citiri, după o prealabilă înscriere (cazul WORM-urilor şi al CD-ROM-urilor).

În cele ce urmează ne propunem o trecere în revistă a tuturor sistemelor menţionate, evidenţiind avantajele şi dezavantajele aferente utilizării fiecăruia precum şi domeniile de aplicabilitate specifice. Excludem în mod intenţionat din analiză discul flexibil deoarece, cu toate avansurile tehnologice actuale (considerabile de altfel), se dovedeşte neadecvat atît ca viteză cît mai ales d.p.d.v. al capacităţii de stocare, sistemelor de arhivare masivă a datelor.

Discul dur (Winchester) - HD

Discul dur prezintă astăzi avantajul unei tehnologii deja mature, deci foarte bine pusă la punct, care îi conferă un raport cost-performanţă foarte bun. Capacitatea de stocare şi timpul mediu de acces la informaţia stocată sînt în continuare creştere, valorile tipice situîndu-se astăzi în gama 20-320 MB respectiv 40-0.5 ms. Un avantaj deloc neglijabil îl reprezintă gabaritul în continuă scădere al acestor dispozitive. "Împachetări" cu factorul de formă de 5 1/4", 3 1/2" sînt deja un standard întîlnit în cele mai multe sisteme de tip PC sau staţie de lucru; cota a fost chiar coborîtă în cazul PC-urilor de tip laptop, ajungîndu-se la factori de 2 1/2". Cuplarea la magistrala sistemului-gazdă se face în majoritatea covîrşitoare a cazurilor printr-un controller SCSI.

Comparativ cu mediile optice, costul unui HD este actualmente mult mai scăzut, de regulă nedepăşind valori maxime de 700-800$. Vitezele ridicate de transfer a datelor recomandă HD în aplicaţii care folosesc intensiv fişiere temporare de dimensiuni relativ mici, ca de exemplu: programe de animaţie, baze de date mici, sisteme de memorie virtuală.

Principalul dezavantaj al HD îl reprezintă capacitatea totuşi mică comparativ cu necesarul unui sistem de arhivare masivă (zeci-sute de GB); există şi dezavantajul deloc neglijabil al unei transportabilităţi reduse a mediului propriu-zis, nefiind disponibile, ca în cazul MO sau DAT, aşa-numitele cartuşe (cartridge) purtătoare ale mediului de stocare, cu care să se "alimenteze" unităţile de disc sau bandă - este cunoscut faptul că unitatea de HD nu reprezintă altceva decît un ansamblu de unul sau mai multe discuri magnetice dure, împachetate în aceeaşi carcasă şi solidare pe acelaşi ax central; ansamblul este fixat în carcasa sistemului gazdă (variante de HD transportabile sînt apariţii de ultimă oră şi sînt încă neconfirmate de piaţă).

Discurile optice

Reprezintă o tehnologie nouă, în plină ascensiune, cu salturi calitative spectaculoase de la an la an. După cum s-a menţionat anterior, în această categorie sînt incluse *discurile magneto-optice (MO), discurile WORM şi dispozitivele de tip CD-ROM,* fiind, în esenţă vorba, de două mari clase de astfel de sisteme: discuri cu conţinut alterabil prin operaţii normale de citire/scriere şi discuri cu conţinut (parţial) nealterabil.

Constructiv, în cazul discurilor MO, sistemul include un mediu de înregistrare transportabil (discul optic propriu-zis) şi un dispozitiv de citire/scriere (unitatea de disc MO). Discul MO combină tehnologiile magnetismului şi opticii pentru înregistrarea informaţiei. Mediul de înregistrare (stratul aferent), este realizat din aliaje de metale din grupul pămînturilor rare. Proprietăţile specifice acestora conduc la un foarte stabil proces de regăsire corectă a informaţiei stocate (minim 10 ani).

Înainte de înregistrare, domeniile magnetice din stratul de aliaj sînt foarte magnetizate în aceeaşi direcţie, la unghiuri drepte faţă de suprafaţa mediului. Pentru a putea scrie pe disc, o bobină este astfel alimentată încît cîmpul magnetic generat să fie orientat în sens opus celui al domeniilor magnetice; cînd o rază laser încălzeşte mediul într-o anumită zonă, domeniile magnetice aferente zonei respective îşi schimbă sensul magnetizării căpătînd sensul impus de cîmpul magnetic exterior.

Citirea informaţiei implică mai întîi decuplarea cîmpului magnetic exterior, apoi o rază laser de mică putere se reflectă pe suprafaţa mediului iar lumina reflectată este detectată fotoelectric. Polaritatea domeniilor magnetice determină polarizarea razei reflectate, detectîndu-se astfel acele domenii "înscrise". Ştergerea informaţiei constă în refacerea direcţiei originale a magnetizării domeniilor din aliaj, operaţie complet analogă celei de înregistrare cu deosebirea că sensul cîmpului magnetic exterior va fi opus de această dată (bobina generatoare este alimentată invers).

În cazul WORM (write once, read many times), proprietăţile de transportabilitate se menţin dar unitatea de disc va

fi capabilă de o singură scriere şi în rest doar de citire. Modalitatea de înregistrare a informaţiei pe aceste tipuri de discuri optice este următoarea: o rază laser produce mici găuri, prin ardere, în suprafaţa mediului folosit (un aliaj pe bază de telur); în timpul procesului de citire, raza laser este reflectată difuz în momentul incidenţei pe o zonă arsă anterior, în caz contrar reflexia fiind puternică.

Un detector bazat pe fotodiodă este capabil să distingă între cele două tipuri de reflexii, asociindu-le cu simbolurile logice binecunoscute (0,1). Înregistrarea de tip "write-once", prezentată mai sus, are avantajul unei mari durate de viaţă. Pentru că nu poate fi alterat sau şters, un astfel de mediu poate fi o bună alegere pentru sisteme de arhivare de masă a informaţiilor, care necesită foarte rar modificări (intervale de timp de ordinul anilor).

Cele mai importante avantaje ale discurilor optice sînt sintetizate în cele ce urmează.

Capacitate de stocare foarte mare: tipic, în plaja 320 - 1000 MB per cartuş (mediul optic propriu-zis), de aceea, prezintă o mare utilitate în aplicaţii în care se vehiculează mari volume de date care, la rîndul lor, necesită un spaţiu imens pentru stocare, adică exact în acele situaţii în care HD-urile nu mai pot fi utilizate cu succes.

Discurile optice pot stoca mai multă informaţie în mai puţin spaţiu, comparativ cu HD. Sistemele fiind bazate pe medii transportabile (cartuşe), rezultă că o singură unitate poate fi folosită pentru vehicularea unei cantităţi practic nelimitate de informaţie. Acest avantaj provine, în esenţă, din faptul că tehnologia bazată pe laser este mult mai precisă ca acţiune asupra mediului şi necesită ca atare mai puţin spaţiu pentru tehnologia eminamente magnetică bazată pe tradiţionalele capete de citire/scriere.

La discurile WORM acest avantaj este dublat şi de cel al unei stocări a datelor foarte durabilă (greu alterabilă de factori externi). Spaţiu mai mare de stocare nu înseamnă însă, numai simpla "înghesuire" a mai multor sute de MB de date pe un disc ci, mai mult, tinde să schimbe fundamental însăşi ceea ce se poate face astăzi cu un sistem de tip PC: în loc să se lucreze doar cu părţi disparate ale unui volum foarte mare de date, se poate dispune, graţie folosirii unui disc optic, de întreg volumul într-un singur loc, în acelaşi moment.

De exemplu, dacă se lucrează cu fişiere care conţin imagini mari, un astfel de disc nu numai că permite stocarea tuturor acestor imagini pe acelaşi mediu fizic ci permite şi o manipulare, o comparare şi o actualizare a acestora online, mult mai uşoară comparativ cu alte dispozitive, rezultată şi din accesul direct, rapid la oricare din fişierele respective.

Discurile optice, atît MO cît şi WORM, sînt mediile ideale astăzi pentru stocarea de documente atît de natură grafică (desene, fotografii, imagini de texte) cît şi sub forma de text propriu-zis.

Desigur, se pot enumera şi dezavantaje, în majoritate surmontabile însă. Un prim dezavantaj poate fi considerat gabaritul încă relativ mare al unităţilor tipice de disc optic comparativ cu cel al uneia de HD, majoritatea fiind produse într-un factor de formă de 5 1/4" (şi înălţime mare); totuşi, sînt uşor integrabile în majoritatea sistemelor de calcul. Un al doilea dezavantaj îl poate constitui timpul mediu de acces mai mare comparativ cu cel al unui HD tipic.

Ordinul de mărime este de zeci de ms, o valoare des întîlnită fiind cea de 40 ms, în timp ce standardul la HD-uri a coborît pînă la valori de ordinul a 14-17 ms. Oricum, pentru cele mai multe dintre aplicaţiile în care discurile optice pot fi folosite cu succes, diferenţa de mai sus nu este deloc semnificativă. Se poate adăuga că pierderea de viteză faţă de HD, se manifestă la discurile optice în special la scriere (este aici cazul discurilor magneto-optice), proces care implică o ştergere prealabilă a zonei în care urmează să se înregistreze noua informaţie.

În sfîrşit, un ultim dezavantaj îl constituie preţul încă ridicat al unei unităţi de disc tipice situat astăzi peste pragul de $3500. O dată achiziţionată o unitate de disc, vor fi necesare în continuare doar cartuşele conţinînd mediul propriu-zis, evaluate la $100 - $200 per bucată. Rezultă că pe termen lung un disc optic poate fi considerat mai avantajos d.p.d.v. financiar decît un HD.

Dincolo de bariera sutelor de MB oferiţi de un singur disc optic exploatat în unitatea aferentă, se poate uşor pătrunde apelînd la aşa-numitele biblioteci de discuri optice (optical disk library). Acestea reprezintă sisteme complexe de stocare funcţional asemănătoare cu tonomatele muzicale în sensul în care permit exploatarea unui set de zeci pînă la sute de discuri magneto-optice, de 12" sau 5 1/4", cu capacitate tipică de 1-2 GB fiecare, în regim complet automat; accesul la fiecare din discuri se realizează graţie înglobării unui subsistem mecanic complex, controlat de regulă, pe bază de microprocesor de 32 biţi, care va executa în condiţii optime de viteză şi precizie transferul discului solicitat din magazia de stocare aferentă în unitatea de disc şi invers.

Capacitatea de stocare totală online pusă la dispoziţia utilizatorului devine astfel de zeci pînă la sute de GB, ceea ce, coroborat cu timpul mediu de acces la un disc din magazie de ordinul a 6-7 secunde, cu timpul mediu de acces la informaţia de pe disc, în plaja 90-150 ms, cu rata foarte redusă a erorii la citire (max. 10E - 12), şi în sfîrşit, cu avantajul incontestabil al accesului direct la informaţie, ar putea conduce la concluzia deloc lipsită de temei, că bibliotecile de discuri optice sînt ideale pentru implementarea unor sisteme puternice de arhivare masivă a imaginilor (text şi desen). În realitate performanţele, remarcabile de altfel, sînt contrabalansate de preţuri încă mult prea ridicate.

Dispozitivul de tip CD-ROM reprezintă o variantă perfecţionată a compact-disc-ului audio, sistem larg răspîndit începînd cu anii '80 pe piaţa mondială a echipamentelor de redare a muzicii de înaltă fidelitate. În esenţă, sistemul este bazat tot pe un disc optic care însă permite doar citiri ale informaţiei (ROM = Read Only Memory) bazate tot pe tehnologia laser. Spre deosebire de discurile WORM, înscrierea informaţiei nu se face de către utilizator utilizînd propria unitate de disc ci de către producător, în procesul de fabricaţie. În privinţa principiilor de înregistrare a informaţiei, a modului de citire cît şi a capacităţii de stocare (tipic, sute de MB) asemănările cu WORM sînt însă evidente. În privinţa vitezei, se poate afirma că discurile CD-ROM sînt mai lente comparativ cu HD-urile şi discurile MO datorită tehnicii CLV (Constant Linear Velocity) de înregistrare, care constă în organizarea informaţiei sub

forma unei piste spirale continue, citită secvenţial.

Banda magnetică digitală audio (DAT)

Sistemele de stocare bazate pe DAT (Digital Audio Tape) au apărut în 1988, introduse de firma GigaTrend; de la această dată şi pînă în prezent, numeroşi alţi producători de prestigiu incluzînd firmele Sony, JVC, Hitachi, Hewlett-Packard s.a. s-au raliat noii tehnologii.

Sistemele DAT se bazează practic pe tehnologia de înregistrare digitală pe bandă magnetică folosită pe scară largă în anii '80 pentru imprimări muzicale de înaltă fidelitate. Mediul de înregistrare îl constituie banda magnetică de 4 sau 8 mm, plasată în casete asemănătoare (dar mult mai mici) cu cele folsite în sistemele video VHS. Capacitatea de înregistrare tipică a unei astfel de casete variază în plaja 1 - 2.5 GB. Citirea şi scrierea informaţiei se realizează în unităţi de bandă specializate, livrate atît intern (adică montate în carcasele calculatoarelor) cît şi extern. Factorul de formă este în ambele cazuri compatibil cu cel al unităţilor de FD sau HD (tipic 5 1/4").

Comparativ cu sistemele clasice de stocare pe bandă magnetică (de exemplu, benzile de 1/4" sau de 8 mm), DAT prezintă incontestabile avantaje, rezumate mai jos:

Caracteristica	Banda de 1/4"	DAT
1. Capacitate (actuală)	60 - 500 MB	2,5 GB
2. Capacitate (prev.)	1.0 GB	5.0 GB
3. Densitate de înreg.	12500 bpi	61000 bpi
4. Rata max. de transfer	90 KB/s	207 KB/s
5. Viteza benzii	90 "/s	0.32 "/s
6. Metoda de înregistrare	analogică	digitală
7. Costul mediu per 100 MB	$25	$1.20

Deşi cele două tehnologii coexistă încă, din tabelul prezentat rezultă un raport cost-performanţă net favorabil sistemelor DAT.

Una dintre cheile acestor performanţe o reprezintă faptul că unităţile de bandă DAT folosesc capete magnetice de citire/scriere de tip rotativ (plasate pe un tambur), sistem întîlnit în aparatura video, rezultînd o scanare a benzii de tip elicoidal; de aici, densitate mult mai mare de înregistrare a informaţiei, lipsa necesităţii unei viteze de defilare ridicate a benzii şi deci, solicitări mecanice reduse, toate comparativ cu sistemele clasice de stocare pe banda magnetică.

De exemplu, o bandă exploatată într-un sistem clasic, este garantată în mod tipic la cca. 400 de treceri - este cazul cartuşelor de 60 MB - în timp ce o casetă DAT de 4 mm permite peste 1.000 de treceri complete prin faţa capetelor de citire/scriere; de asemenea, într-o singură trecere, un sistem DAT poate înregistra 1 GB de date în timp ce tehnologia clasică a benzilor de 1/4" impune 24 de treceri pentrù înregistrarea a doar 150 MB.

Sistemele DAT necesită, datorită capacităţii imense de stocare pe care o deţin, tehnici perfecţionate de detecţie şi corecţie a erorilor; pentru o bandă de 1,2 GB de exemplu, rata convenţională de detecţie a erorilor de 1/10E8 nu este nici pe departe suficientă. Ca atare, sistemele DAT fac apel la noua tehnică numită FEC (Forward Error Correction) care permite reducerea ratei la valoarea de 1/10E15, adică, practic, o eroare se poate constata la 1.000.000 benzi distincte.

Sistemele DAT permit trei moduri distincte de acces la informaţia stocată: streaming, aleator şi actualizare-pe-loc (update-in-place). Modul streaming serveşte la adăugarea de noi date pe bandă: pentru modificarea unui fişier deja stocat, pe bandă se va scrie în întregime noua sa versiune, toate versiunile precedente ale fişierului respectiv rămînînd şi ele pe bandă.

În modul de acces aleator, orice fişier de pe o bandă de 1-2 GB poate fi localizat în mai puţin de 60 secunde; în acest mod sistemele DAT emulează un dispozitiv tipic specializat în lucrul cu volume de date organizate pe blocuri, ca de exemplu FD sau HD. Modul actualizare-pe-loc reprezintă cea mai avansată tehnică de acces aleator la informaţie; în acest caz, accesul la date se realizează printr-o combinaţie între metodele aleatoare şi secvenţiale, existînd facilităţi de adăugare (append) cît şi de suprascriere (overwrite), menţinîndu-se totodată şi caracterul compact al întregului volum de date.

Unul dintre puţinele dezavantaje ale sistemelor DAT îl reprezintă, desigur, viteza mică în modul de acces aleator la fişiere comparativ cu sistemele bazate pe tehnologia optică (WORM, CD-ROM, discurile magneto-optice) sau cea magnetică alternativă (HD). Dar costul unui disc optic este mai mare decît cel al unei casete DAT de 4 sau 8 mm, iar capacitatea de stocare inferioară (tipic, 600-800 MB).

Date fiind caracteristicile enunţate cît şi faptul că sistemele bazate pe disc optic cunosc în prezent o puternică dezvoltare, DAT pare a fi sistemul ideal (rapid şi fiabil) pentru păstrarea copiilor de salvare (backup) a informaţiei curent manipulate pe discurile optice. În acest sens, o aplicaţie tipică constă în asocierea unui sistem DAT cu server-ul unei reţele LAN cu rolul de arhivare şi backup a informaţiei comune necesară tuturor utilizatorilor cuplaţi în reţea.

În final...

... se poate spune că în momentul de faţă, este greu de estimat care dintre sistemele prezentate va cîştiga competiţia, progrese spectaculoase înregistrîndu-se în fiecare an atît în sfera dispozitivelor magnetice cît şi în cea a dispozitivelor optice (de exemplu, firma Maxstorage a lansat HD-ul de 2GB iar sistemele de stocare holografică în cristal sînt pe punctul de a depăşi faza de experiment). Cel mai probabil este că, cel puţin în viitorul previzibil, ambele tehnologii vor coexista completîndu-se reciproc.

BIBLIOGRAFIE

- 1. BYTE Magazine, noiembrie 1990, p. 275-327
- 2. Computer Product News, noiembrie 1990, p. 5-6

ing. Dan Onţanu

scop

LIDERUL REȚELELOR DE CALCULATOARE ÎN ROMÂNIA A ÎNFIINȚAT PRIMUL CENTRU NAȚIONAL AUTORIZAT DE INSTRUIRE

NOVELL®

- ☐ Instructori atestaţi oficial - NOVELL CERTIFIED INSTRUCTORS -
- ☐ Cursuri cu o durată de două săptămîni în condiţii de pregătire ireproşabile pe o reţea de calculatoare de cea mai înaltă performanţă
- ☐ La cerere cursurile se pot desfăşura în limbile engleză sau franceză
- ☐ Participanţii primesc diplome originale NOVELL prin care se atestă absolvirea cursului

Novell Authorized Education Center

Pentru înscrieri telefonaţi urgent la tel: 11.74.21; 11.92.48, fax: 11.73.74

scop

Noi sîntem soluţia!

Super-floppy

Dischetele de 3,5" revin în forţă mai bune şi avînd o capacitate mai mare decît niciodată. Unitatea de 2,8 MB produsă de IBM pare să fi cîştigat, dar modele mult mai interesante de 20 MB, ale unor producători mai mici, pot impune noi standarde.

Sîntem martorii unei adevărate revoluţii în ceea ce priveşte mediile de stocare şi de backup (salvare/restaurare) pentru calculatoarele personale. În centrul atenţiei, pe lîngă unităţile de discuri optice pentru scriere/citire şi unităţile de bandă magnetică de ordinul gigaocteţilor se află şi dischetele de 3,5" de capacităţi mărite.

Să vedem care a fost desfăşurarea evenimentelor. Vara trecută IBM anunţa o unitate de 2,8 MB produsă de Toshiba Corp. Întreprinderea Insite Peripherals Inc. din San Jose a început să-şi comercializeze unităţile de dischete de 20 MB denumite Floptical drive. Brier Technology Inc., o altă întreprindere din San Jose, îşi vinde propriile unităţi de 20 MB, denumite Flextra drive, începînd din primăvara trecută.

Cît de semnificative sînt aceste superfloppy? Mulţi analişti sînt de părere că unitatea de 2,8 MB a IBM-ului reprezintă o evoluţie. La fel ca unitatea de 3,5" de 1,4 MB dinaintea ei, unitatea de 2,8 MB nu face decît să dubleze capacitatea predecesoarei unităţi. Unitatea, care utilizează dischete pe bază de ferită cu bariu, este considerată ca următorul pas logic în domeniul dischetelor de 3,5". Introdusă în iunie, unitatea IBM-ului, face parte din configuraţia standard a calculatoarelor PS/2 model 57 şi este livrată opţional la modelele 35 şi 40 şi se aşteaptă ca unitatea să impună un nou standard.

Unităţile de 3,5" cu o capacitate de 20 MB reprezintă mai mult o revoluţie în domeniu decît o evoluţie. Datorită capacităţii lor ridicate de stocare, ele permit utilizatorului să-şi gestioneze datele mult mai eficient decît cu ajutorul dischetelor obişnuite.

Cei doi ofertanţi de superfloppy de 20 MB, Insite Peripherals şi Brier Technology, sînt două nume cvasi-necunoscute, cel puţin pînă acum. Ambele întreprinderi îşi comercializează unităţile prin intermediul OEM (Original Equipment Manufacturers), iar preţurile acestora sînt cuprinse între 400 şi 1000$.

Din momentul din care IBM-ul şi-a anunţat noua unitate de 2,8 MB, mai multe companii au anunţat producerea de suport pentru ea.

Sony Corp. şi Toshiba, de ex., produc atît unităţi cît şi dischete care utilizează noul format. Alte companii au început să producă totul pentru aceste unităţi, de la software pentru backup pînă la controller-e de disc.

Şi Insite şi Brier par să cîştige suport pentru formatele lor, mult mai exotice, de 20 MB. În luna iunie la expoziţia PC Expo din New York, firmele Insite, Iomega Corp., Maxell Corp. şi Minnesota Mining and Manufacturing Co. (3M) au înfiinţat asociaţia Floptical Technology Association care îşi va avea sediul în

Doar pentru că se pot aduna mai multe informaţii pe o dischetă de 3 1/2" nu înseamnă că va trebui să renunţăm la fiabilitate. De fapt, noua generaţie a floppy-urilor de mare capacitate este mult mai fiabilă decît driver-ele mai vechi, datorită tehnicii sotisficate de corectare a erorilor înglobată în ele.

Toate unităţile noi ale firmelor Brier Technology Inc., IBM şi Insite Peripherals Inc folosesc discuri din ferită cu bariu în locul discurilor din oxid feros cu cobalt gama, avînd capacităţi de 720 KByte - 1,4 MByte. Ferita cu bariu poate înmagazina celule magnetizate mai mici şi cu o densitate mai mare de informaţii.

Unităţile de 2,8 MB folosesc o înregistrare mult mai eficientă, perpendiculară, în care celulele sînt magnetizate vertical în loc de orizontal.

Simpla migrare la un nou tip de floppy-disc ajunge pentru IBM. Unitatea sa de 2,8 MB conţine de două ori mai multe sectoare în acelaşi număr de piste, dublînd capacitatea discului de la 1,4 la 2,8 MB.

Pentru Inside şi Brier, producătorii unităţilor de 20 MB, ferita cu bariu reprezintă doar jumătate din cîştig. Produsele lor au mult mai multe piste decît cele ale firmei IBM, deci ei vor trebui să poziţioneze cu o mai mare precizie capetele de scriere/citire. Acest lucru înseamnă o regîndire fundamentală a mecanismului unităţii.

Ambele formate de 20 MB folosesc pentru poziţionarea capetelor servosisteme, asemănătoare celor aflate pe hard-disk-uri şi pe discurile optice. Insite foloseşte piste optice întreţesute în materialul de înregistrare de unde şi numele floptical. Brier foloseşte o schemă de înregistrare dual-magnetică pentru a depune un strat de servo-piste cu un curent mai mare şi o frecvenţă mai mică decît a pistelor de date, în momentul producerii discului. Servopistele sînt invizibile pentru bobina de date din capul de citire, dar sînt citite de cea de-a doua bobină, cea pentru servopoziţionare.

În ciuda densităţilor, discurile floppy de 20 MB ar trebui să-şi dovedească fiabilitatea. Motivul: Atît Brier cît şi Insite folosesc o amănunţită metodă de detectare a erorilor şi scheme de corecţie.

Vicepreşedintele firmei Insite, Chuck Moran pretinde că unitatea sa poate corecta erori de pînă la o lungime de 80 biţi. Şi unitatea Brier are facilităţi similare de corectare a erorilor. Ambele folosesc piste goale în care rescriu în mod automat informaţia din domeniile avariate.

Cupertino, California, şi care îşi propune ca scop promovarea unităţilor şi dischetelor Floptical. În acelaşi timp mai mulţi producători importnanţi de produse SCSI (small computer systems interface) au anunţat că vor oferi suport pentru noul format. Pentru a nu rămîne pe dinafară, Brier încearcă de asemenea să-şi asigure suport pentru dischetele Flextra. Quadram vinde unităţile Flextra sub numele de QuadFlextra, iar Verbatim Corp. produce dischete.

Nici unul din aceste sisteme nu va putea înlocui pînă la urmă harddisk-urile, pe de o parte din cauza timpilor mici de acces, iar pe de altă parte din cauza capacităţii de stocare limitată. Unitatea IBM, de ex., este de cca. două ori mai rapidă decît o unitate de 1,44 MB, dar cu toate acestea este lentă comparativ cu standardele harddisk-urilor. Unitatea produsă de Insite are un timp de acces de cca. 80 ms - cam acesta fiind timpul de acces al primelor generaţii de harddisk-uri. Unitatea produsă de Brier este cea mai rapidă, ea ajunge la un timp de acces de cca. 35 ms, valoare mult mai apropiată de timpul de acces al harddisk-urilor moderne, care este în jur de 25 ms.

Aceste superfloppy îşi vor putea găsi o primă aplicare ca medii de salvare/restaurare (backup). A salva un harddisk de 80 MB pe dischete de 1,44 MB nu este chiar o plăcere. Pe dischete de 20 MB este altceva. Pe dischetele de 2,8 MB ale IBM-ului, numărul dischetelor necesare pentru backup se reduce la jumătate, şi deoarece aceste unităţi sînt de cca. 2 ori mai rapide decît cele de 1,44 MB şi timpul de salvare se înjumătăţeşte.

Desigur este mai convenabil un backup pe bandă magnetică decît pe dischete, fie ele chiar de 20 MB. Numai că fiecare PC are cel puţin un lector de dischete, în timp ce o unitate de bandă (streamer) rămîne încă un lux.

O altă utilizare a superfloppy-urilor ar putea-o constitui stocarea pe termen lung a datelor şi programelor care nu sînt utilizate frecvent. Pe aceste dischete s-ar putea salva subdirectoare întregi şi la nevoie s-ar putea restaura sau s-ar putea lansa în execuţie anumite programe direct de pe ele.

Unităţile IBM de 2,8 MB pot constitui o alternativă pentru obişnuitele unităţi de 1,44 MB şi în ceea ce priveşte costurile. Unităţile sînt doar cu ceva mai scumpe decît cele de 1,44 MB, iar preţurile dischetelor sînt similare, dacă ţinem cont de capacitatea de stocare suplimentară. Aceste unităţi sînt 100% compatibile cu alte unităţi de 3,5".

Modelele de 20 MB costă mult mai mult decît lectoarele de dischete existente şi nici dischetele nu sînt ieftine (20-25 bucata).

Totuşi ţinînd cont de anumite considerente, utilizarea lor se poate dovedi raţională. Astfel, între anumite limite, ele pot înlocui atît lectorul de dischete obişnuit cît şi o unitate de salvare pe bandă magnetică. Pentru aceasta însă ele ar trebui să fie compatibile cu unităţile convenţionale atît de 720 KB cît şi de 1,44 MB. Dar Insite şi Brier au ales soluţii diferite de asigurare a compatibilităţii.

Unitatea Floptical produsă de Insite poate scrie şi citi dischete de 720 KB şi de 1,44 MB dar nu şi dischetele de 2,8 MB. Unitatea produsă de Brier nu este compatibilă în scriere/citire cu formatele anterioare. Brier insistă asupra faptului că utilizatorii sînt mult mai interesaţi de capacităţi de stocare mărite decît de compatibilitate. Analiştii se aşteaptă ca Brier să anunţe în 1991 unităţi cu capacităţi de pînă la 50 MB, compatibile cu modelele proprii anterioare.

Din cauză că toate aceste unităţi de dischete utilizează sistemul de operare MS-DOS, toate sînt suportate, într-o formă sau alta, de soft-urile de backup.

Cîteva pachete de programe pentru backup merg dincolo de limitele DOS şi utilizează formatul propriu - de regulă mai rapid - al acestor unităţi. Sytron Corp., producătoarea soft-ului pentru backup Systos Plus, suportă direct formatul de 2,8 MB. Gazelle Systems Inc. suportă de asemenea formatul de 2,8 MB cu soft-ul său Back-It 4.

Fifth Generation Systems Inc. nu suportă nici unul dintre formatele de mare capacitate. Dar, ca în cazul multor produse pentru backup, Fastback, produsul acestei companii, permite o setare implicită care să trateze orice unitate DOS ca un harddisk. Tehnica este cu ceva mai lentă deoarece nu oferă acces direct la memorie (DMA), dar în versiunile ulterioare va fi oferit suport şi pentrru aceste unităţi. Nici PC Tools 7.0, produs al firmei Central Point Software Inc., nu suportă nici un format direct, decît prin intermediul DOS. Compania a anunţat însă că va oferi suport în versiunile următoare.

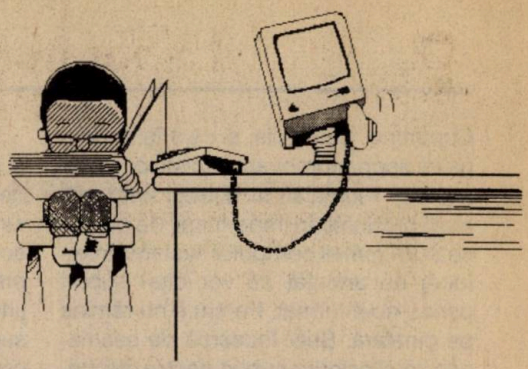
Unitatea de 2,8 MB se va impune aproape sigur dacă ţinem cont de costul relativ redus, de compatibilitatea deplină cu versiunile anterioare şi de faptul că este promovată de IBM. A răspunde cine va fi cîştigătorul - dacă va fi vreunul - în cazul dischetelor de 20 MB este greu. Brier a fost prima întreprindere care a comercializat un astfel de produs. Dar datorită suportului oferit de producătorii de controller-e şi de dischete, Insite i-a luat-o probabil înainte.

Un alt aspect critic îl va constitui suportul oferit de producătorii de PC-uri. Dacă mai mulţi vînzători vor începe să-şi echipeze calculatoarele incluzînd în configuraţia standard unităţi de 20 MB probabil că ceilalţi îi vor urma.

(R.M.)

Mica publicitate

- Ofer documentaţie VMS / RSX, eventual la schimb. Tel. 90/233808.
- Caut soft pentru Amiga. Tel. 934/34128 Cătălin 18-22.
- Cumpăr documentaţie, programe, jocuri, command module, accesorii, pentru Texas Instruments TI-99-4A computer. Stănculescu Marin, str. Ciuruleasa nr. 3, sect. 4, cod 75445, O.P. 7, Bucureşti, tel. 864588.
- Vînd sau schimb o gamă variată de programe utilitare pentru calculatoare compatibile IBM-PC. Tel. 167813.



Anatomia fişierelor PCX

Formatul PCX a fost inventat de către compania americană Zsoft. Această companie a lansat programul de grafică denumit PC Paintbrush. O versiune a acestui program este livrată cu produsul firmei Microsoft, Windows.

În funcţie de modul de afişare pe ecran pentru care au fost destinate, fişierele PCX sînt împărţite în două sau trei părţi:

- un bloc de informaţii de 128 de octeţi numit antet;
- un bloc de date grafice.
- pentru modul 19 de afişare (adică 320x200 în 256 de culori valabil pentru VGA sau MCGA) există o a treia secţiune situată după celelalte două, conţinînd informaţii despre paleta de culori; fiecare din cele 256 de culori afişabile putînd fi alese dintr-o multitudine de culori furnizate de către placa VGA, in acest caz fişierul trebuind să indice care culori au fost selectate.

Fişierele PCX pentru celelalte moduri de afişare trebuie de asemenea să conţină informaţii de culoare, dar deoarece sînt cel mult 16 culori afişabile, volumul de date necesare pentru a le descrie încape în antet.

Pentru afişarea unui fişier PCX trebuie citite în primul rînd informaţii din antet, ca de exemplu setarea paletei, apoi trebuie decodificate datele grafice, după care acestea vor fi afişate pe ecran. Antetul este împărţit în cîteva cîmpuri, structura sa fiind arătată în tabelul următor:

Imaginea de pe ecranul calculatorului este compusă dintr-o multitudine de puncte numite pixeli, care sînt aranjate în coloane şi rînduri. Numărul de pixeli din rînduri sau coloane, precum şi numărul de culori posibile pe care aceştia le pot afişa depinde de modul de afişare. De exemplu modul 16, valabil pentru EGA sau VGA, este 640 coloane x 350 rînduri x 16 culori posibile. Modul 18, valabil numai pentru VGA, este 640 coloane x 480 rînduri x 16 culori.

Numele cîmpului	Octeti	Descriere
Cod	1	Întotdeauna 10
Versiune	1	0 = v 2.5
		2 = v 2.8 cu informaţie despre paletă
		3 = v 2.8 fără informaţie despre paletă
		5 = v 3.0 cu informaţie despre paletă
Comprimare	1	1 = comprimat
Biţi pe pixel	1	1 pentru monocrom, EGA modul 16, VGA modul 18
		8 pentru MCGA/VGA modul 19
Coordonată X minimă	2	întreg de 16 biţi
Coordonată Y minimă	2	întreg de 16 biţi
Coordonată X maximă	2	întreg de 16 biţi
Coordonată Y maximă	2	întreg de 16 biţi
Rezoluţie orizontală	2	dimensiunea ecranului în pixeli
Rezoluţie verticală	2	
Paletă	48	16 x 3 octeţi pentru R, G, B.
Rezervat	1	
Plane	1	4 pentru modurile 16 şi 18, 1 pentru modul 19
		modul monocrom şi celelalte nu necesită utilizarea planelor multiple
Octeţi pe linie	2	80 pentru modurile 16, 18, 19 - ecrane întregi

Felul în care imaginea este stocată în memoria video depinde de asemenea de mod. Modul 19, 320x200x256 culori, este stocat într- un bloc continuu de memorie de 64000 octeţi. Fiecare octet deţine informaţia pentru un pixel, de aici limita de 256 de culori.

Modurile 16 şi 18 sînt aproape identice, deosebirea consta în faptul că în modul 18 imaginile sînt mai lungi (480 de rinduri faţă de 350 în modul 16). În continuare vor fi descrise aceste moduri, după înţelegerea cărora modul 19 şi modul 7 (monocrom) sînt deductibile. De asemenea programul Windows al firmei Microsoft foloseşte tôate aceste moduri.

Pentru simplitate se va descrie modul 16; pentru modul 18 trebuie avută în vedere doar diferenţa de rînduri. Structura datelor grafice in fişierul PCX în modul 16 reflectă felul în care acestea sînt stocate în memoria video.

Avînd 16 culori înseamnă că fiecare pixel poate fi descris de către un număr binar de 4 biţi: 0000 este culoarea 0, 0001 este culoarea 1, 0010 este culoarea 2, şi aşa mai departe pînă la 1111 care este culoarea 15.

Un monitor color contine trei tunuri de electroni - unul pentru culoarea roşie(R), unul pentru verde(G) şi altul pentru culoarea albastră(B). Culoarea fiecărui pixel este formată dintr-o combinaţie de roşu, verde şi albastru. Semnanul de la placa video la monitor conţine informaţii despre componentele RGB a fiecărui pixel, plus o componentă de intensitate. Dacă fiecare tun RGB poate fi activ sau inactiv, pot exista opt combinaţii de culori posibile. Dacă aceste culori pot fi slabe sau intense, posibilităţile se dublează la 16. Pentru fiecare pixel cele patru componente ale semnalului sînt controlate fiecare de către un bit într-un număr descriptiv stocat în memoria video. Cel mai puţin semnificativ bit, cel din dreapta numărului, este pentru albastru, bitul 1 pentru verde, 2 pentru roşu, iar al patrulea este pentru intensitate.

Un bit avînd valoarea '1' înseamnă tunul respectiv activ, '0' inactiv. Acelaşi lucru se aplică şi intensităţii.

Deci fiecare octet din memoria video conţine informaţii despre doi pixeli. Problema este că un ecran de 640x350x4 biţi necesită 112000 octeţi. Secţiunea din harta de memorie a calculatorului alocată ecranului a fost fixată mai demult cînd cerinţele erau modeste, iar un volum de memorie atît de mare s-ar suprapune peste memoria alocată altor lucruri.

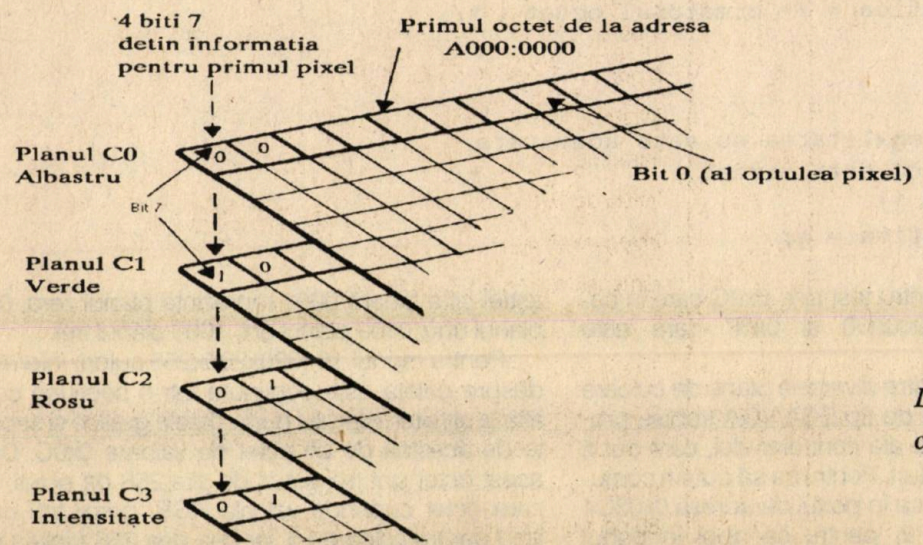
Pentru a înghesui volumul de 112000 octeţi într-un loc mai îngust, aceasta a fost împarţit în patru secţiuni asimilate unor straturi (se poate face o analogie cu filele unei cărţi). Cele patru secţiuni au aceeaşi adresă.

Pentru a citi/scrie din/într-un strat anume trebuie comutat pe acesta, celelalte trebuind ascunse (analogie cu răsfoirea filelor unei cărţi).

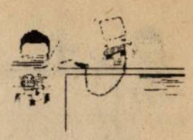
Modul 16/18 alocă straturile, denumite şi plane, componentelor semnalului video. Cei patru biţi ai unui pixel sînt răspindiţi între plane (vezi figura). Deoarece un strat are un bit pe pixel şi sînt 640 de-a lungul ecranului, fiecare linie ocupă 640/8=80 octeţi în fiecare din cele patru plane.

Intuitiv planul albastru este o matrice de 80x350. Fiecare rînd conţine informaţii pentru o linie, şi un element din rînd stochează componenta albastră pentru opt pixeli consecutivi. Deci elementul [0] [0] se referă la primii opt pixeli din stînga sus a ecranului.

Dacă valoarea elementului [0] [0] este zero, nici unul din acesti pixeli nu are o componentă albastră. Dacă valoarea este 00011000 însemnă că pixelii al patrulea şi al cincelea au o componentă albastră.



Dispunerea informaţiei pe cele patru plane de culoare.



Planele roşu, verde, intensitate lucrează pe acelaşi principiu, cu cel mai semnificativ bit din [0] [0] referindu-se întotdeauna la pixelul din stînga sus. În limbajul C fiecare din cele patru plane poate fi accesat printr-un pointer care stochează adresa de memorie a primului element. Toate cele patru matrici încep la aceeaşi adresă, deci cei patru pointeri ar trebui să aibă aceeaşi valoare.

Înainte de a citi sau scrie oricare din matrici, programul trebuie să comute în planul corespunzător, în zona de memorie către care indică pointer-ul.

În practică, însă, nu trebuie să avem patru pointeri deoarece ei au aceeaşi valoare. Avem nevoie doar de un pointer şi de comenzile corespunzătoare de a comuta între diverse plane.

În fişierele PCX (modul 16) straturile roşu, verde, albastru şi de intensitate nu sînt unul după celălalt, ci fiecare linie de ecran este stocată separat. Pe o imagine de mărimea ecranului - în contradicţie cu o imagine mai mică decît ecranul - vor fi 80 de octeţi pentru stratul albastru urmaţi de 80 de octeţi de verde, 80 de roşu şi 80 de intensitate. Împreună aceşti 80x4 octeţi descriu prima linie de ecran. Urmează alţi 80x4 octeţi pentru a doua linie şi aşa mai departe. Singura complicaţie este faptul că datele sînt comprimate pentru a salva spaţiu. Comprimarea este simplă, de tip RLE (run length encoding). De exemplu, într-un fişier unde există 300 de octeţi de aceeaşi valoare ("1" de exemplu), această secvenţă poate fi inlocuită de o secvenţă compusă dintr-un 300 urmată de un 1, precedată de un fanion care indică faptul că următorii cîţiva octeţi reprezintă date comprimate care trebuie expandate.

Un program care citeşte fişierul va vedea fanionul, va citi 300 şi "1" şi va folosi 300 ca şi contor într-o buclă pentru a regenera secvenţa. Dacă fanionul ocupă un octet, 300 ocupă doi octeţi, "1" încă un octet, metoda de comprimare înlocuieşte 300 de octeţi cu patru. Compresia în fişierele PCX este o variantă a acestei scheme. Comprimarea este restrînsă la datele pentru o linie. Deci dacă două linii de ecran adiacente sînt "0" atunci vor fi de două ori 4x80 blocuri de zero. Comprimarea se va opri la sfîrşitul primilor 320 de octeţi şi va reîncepe la a doua secvenţă de 320 de octeţi.

Deci chiar dacă fişierul este comprimat fiecare linie este o entitate separată. Pentru aceeaşi linie, pentru planuri de culoare diferită însă adiacente se poate aplica compresia dacă octetul de la sfîrşitul unuia şi cel de la începutul celuilalt au aceeaşi valoare.

Tehnica de decodificare este urmatoărea :

- se citeşte primul octet de date grafice; presupunem că valoarea lui este g;
- dacă primii doi biţi din g sînt "1", valoarea contorului este egală cu valoarea cuprinsă în următorii şase biţi, iar data grafică comprimată este în octetul ce urmează după g;
- dacă din primii cei doi biţi din g unul sau doi sînt "0" înseamnă că ceea ce urmează după g nu este comprimat, iar valoarea contorului este egală cu "1".

Putem testa şi masca biţii folosind operatorul & (ŞI binar) :

```
if (g & 0xC0 == 0xC0)
    {
    /* daca egalitatea este adevarata
    inseamna ca octetul este marcat     */
    contor = g & 3F;
    data_grafica = /* urmatorul octet   */
    }
else
    {
    /* daca egalitatea nu este adevarata
    octetul nu este marcat              */
    contor = 1;
    data_grafica = g;
    }
```

Măştile folosite pentru test sînt: 0x0C care în codul binar este 11000000 şi 0x3F care este 00111111.

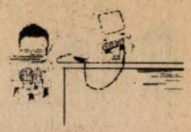
Pentru a comuta între diversele plane de culoare într-un adaptor grafic de tip EGA/VGA trebuie programate nişte registre ale controler-ului, care pot fi accesate ca nişte porturi. Pentru ca să putem comuta între plane trebuie ca în portul de adresă 0x03C4 să punem valoarea 2, pentru ca apoi în portul 0x03C5 să punem numărul planului în care lucrăm; astfel cifra binară 0001 reprezintă planul zero, 0010 planul unu, 0100 planul doi, 1000 planul trei.

Pentru modul 19 (320x200x256 culori) informatia despre paleta este continuta intr-o sectiune ce se afla la sfirsitul fisierului dupa datele grafice si separate de acestea de un octet de valoare 0x0C. Dupa acest octet sint trei seturi de cite 256 de octeti. Fiecare octet cuprinde valorile RGB, primii trei octeti fiind pentru culoarea 0, iar cea dea 256 tripleta fiind pentru culoarea 255. Pentru a face ca imaginea sa

apara instantaneu pe ecran, se seteaza intreaga paleta pe negru si apoi se inscriu datele grafice in memoria video. Se verifica urmatorul octet si daca acesta este 0x0C se indeparteaza. Ultima sectiune a fisierului se citeste intr-un tablou de 256x3=768 octeti. Informatia despre paleta nu este comprimata. Pentru setarea registrilor de culoare se foloseste intreruperea 0x10 cu subfunctia 0x12. Deci AH trebuie sa contina 0x10, AL 0x12, BX contine primul registru de culoare ce trebie setat, iar CX contine numarul de registrii (256) iar ES:DX segmentul respectiv offsetul tabelei de culori. Segmentul este segmentul de date curent, iar offsetul este adresa inceputului tabloului. Daca culorile sint prea stralucitoare se deplaseaza bitii din cei 768 de octeti spre dreapta cu 2.

Programul propus foloseste memoria ecran ca pe o matrice unidimensională unde elementele 0-79 reprezintă primul rînd, 80- 159 reprezintă al doilea rînd ş.a.m.d., şi a fost optimizat din punct de vedere al vitezei. Viteza de afişare va depinde de timpul mediu de acces al suportului magnetic de pe care se face citirea fişierului PCX. Fişierul PCX va trebui să aibă exact dimensiunile ecranului (640x350 în modul 16 sau 640x480 în modul 18).

```
#include <stdio.h>
#include <dos.h>
main(int argc, char *argv[])
{
    FILE *fisier;
    char far *inceput_ecran, far *sfirsit_ecran;
    char antet[128], buffer_fisier[250];
    union REGS regs;
    int i, j, t, paleta, plan_culoare, contor, valoare, offset_linie;
    int pnt_buf_fis;
    static int masca[] = {0, 8, 1, 9};
    if (argc != 2)
    {
        printf("Utilizare : PCXVIEW numefisier.PCX ");
        exit();
    }
    if ( ! (fisier = fopen (argv[1], "rb")))
    {
        printf("Acest fisier nu poate fi deschis");
        exit();
    }
    if ( fread(antet, 128, 1, fisier) != 1 || antet[0] != 10 )
    {
        printf("Antetul nu poate fi citit");
        exit();
    }
    regs.h.ah =  0;
    regs.h.al = 16;
    int86(0x10, &regs, &regs); /* trecere in modul 16 (EGA) */
    for (i = 0; i < 16; i++)   /* pentru fiecare culoare */
        {
        for(j = paleta = 0; j < 3; j++) /*pentru fiecare octet din tripleta RGB */
            {
            t = (antet[16+(i*3)+j] & 255) >> 6;
            paleta |= masca[t] << (2 - j);
            }
        regs.h.ah = 0x10;
        regs.h.al = 0;
        regs.h.bh = paleta;
        regs.h.bl = i;
        int86(0x10, &regs, &regs);
        }
    outp(0x03C4, 2); /* selectia planelor de culoare */
    outp(0x03C5, 1); /* planul 1 */
```



```
inceput_ecran = MK_FP(0xA000, 0);
sfirsit_ecran = MK_FP(0xA000, 350*80); /* 480*80 pentru modul 18 (VGA) */
contor = valoare = offset_linie = 0;
plan_culoare = 1;
pnt_buf_fis = 250;
while( inceput_ecran != sfirsit_ecran )
     {
     if (pnt_buf_fis = = 250)
          {
          pnt_buf_fis = 0;
          fread(buffer_fisier, 1, 250, fisier);
          }
     if ( ((contor = buffer_fisier[pnt_buf_fis++]) & 0xC0) = = 0xC0)
          { /* test pentru date comprimate */
          if ( pnt_buf_fis = = 250 )
               {
               pnt_buf_fis = 0;
               fread(buffer_fisier, 1, 250, fisier);
               }
          valoare = buffer_fisier[pnt_buf_fis++];
          contor &= 0x3F;
          }
     else
          { /* este un octet de date grafice */
          valoare = contor;
          contor = 1;
          }
     while( contor- - )
          {
          inceput_ecran[offset_linie] = valoare; /* se scrie valoarea in memoria video */
          if ( ++offset_linie = = 80 ) /* test de sfirsit de linie in planul curent */
               {
               offset_linie = 0; /* se trece la inceputul unei linii noi */
               if ( plan_culoare = = 8 ) /* 1000 in binar */
                    { /* daca da */
                    inceput_ecran += 80;  /* se trece la inceputul unei linii noi */
                    plan_culoare = 1;     /* 0001 in binar */
                    }
               else
                    /* daca nu */
                    plan_culoare <<= 1;   /* se trece la urmatorul plan de culoare */
               outp(0x03C5, plan_culoare); /* selectia planului de culoare */
               }
          }
     }
     fclose(fisier);
     printf("\a");   /* beep */
     getch();   /* se asteapta pentru apasarea unei taste */
     regs.h.ah = 0;
     regs.h.al = 3;
     int86(0x10, &regs, &regs); /* se trece in modul 3 */
     return;
```

ing. Duşa Silviu

SIMPLY THE BEST

mita

COPIATOARE, FAXURI, IMPRIMANTE LASER

mita COPIERS

mita LASER PRINTERS

mita FAXES

IMPORTATOR AUTORIZAT

Lecom S.R.L.

Calea Victoriei 126 fax 12.95.58. tel. 50.64.20; 50.79.94.

Folosirea listelor în limbajul Prolog

În continuarea cursului de programare logică în Turbo-Prolog vă prezentăm cîteva modalităţi de prelucrare a listelor, folosind acelaşi limbaj de programare

Listele sînt structuri de date importante în filosofia limbajelor descriptive. O listă este o generalizare a noţiunii de vector (structură de date prezentă în alte limbaje de programare). În definirea unei liste nu este necesar să se precizeze numărul de elemente componente. Deci lista este o structură dinamică care conţine o înşiruire de elemente într-o anumită ordine, separate prin virgulă şi cuprinse între paranteze pătrate.

Exemplu de listă de întregi:

```
Lista = [24,36,12,11,0,4033,89]
```

-listă de şiruri:

```
L = ["unu","doi","trei","patru"]      etc.
```

Nici unul dintre predicatele predefinite din limbajul Prolog NU operează cu liste în mod explicit. Singura facilitate oferită este aceea de delimitare a primului element din listă de celelalte, prin următoarea convenţie:

- primul element al listei este delimitat de celelalte elemente prin caracterul "|". Putem să rescriem prima listă astfel:

```
Lista = [X|Y]
unde X = 24 iar Y = [36,12,11,0,4033,89].
```

X este denumit generic în literatura anglo-saxonă "**head**", adică în traducere "cap", iar Y "**tail**", adică coadă. Se mai folosesc şi denumirile "antet" şi "corp". În continuare vom folosi primele două expresii. Deci putem afirma că lista "Lista" este compusă din capul listei X şi coada listei Y. O listă poate să nu conţină nici un element, caz în care dispar noţiunile de cap şi coadă. Lista vidă este reprezentată prin simbolul []. Lista vidă va putea fi folosită pentru a pune condiţiile la limită în definirea predicatelor. Pot exista, de asemenea şi liste de liste:

```
L_de_l = [[1,2,3], [3,2,1], [2,3,1], [1,3,2],
          [1,1,1], [2,2,2], [3,3,3]]
(sau liste de liste de liste .....).
```

Deşi aparent nu putem face mare lucru cu listele noastre, aceste convenţii de notaţie ne permit să scriem cîteva predicate care pot face prelucrări deosebit de complexe. Aveţi îndoieli? Experimentaţi programele care urmează. Primul predicat pe care îl vom scrie ne va folosi pentru a concatena două liste.

```
append([],X,X).
append([H|T],List,[H|T2]):-
append(T,List,T2).
```

Să scriem un mic program care să folosească predicatul nostru:

```
DOMAINS
INTEGERLIST = INTEGER*
PREDICATES
append(INTEGERLIST,
       INTEGERLIST,INTEGERLIST)
CLAUSES
append([],X,X).
append([Head|Tail],List,[Head|T2]):
          -append(Tail,List,T2).
GOAL  X = [1,2,3,4,5],Y = [6,7,8,9],
append(X,Y,Z),write(Z).
```

Rezultatul va fi:

```
[1,2,3,4,5,6,7,8,9]
```

În funcţie de modul în care apelăm predicatul "**append**", putem efectua mult mai multe operaţii asupra listelor fără a fi nevoie să intervenim asupra definiţiei iniţiale.

- Dacă A = [1,2,3], B = [4,5,6] şi C = [1,2,3,4,5,6] predicatul append(A,B,C) va raporta "TRUE". În această situaţie putem obţine confirmarea faptului că A concatenat cu B dă lista C.
- Dacă A = [1,2,3] şi C = [1,2,3,4,5,6] predicatul va returna B = [4,5,6] în situaţia în care cerem append(A,B,C).
- Dacă A şi B sînt două variabile libere şi C este legată de valoarea [1,2,3,4,5,6] vom obine toate listele care prin concatenare dau lista C, adică:

```
A = [],B = [1,2,3,4,5,6]
A = [1],B = [2,3,4,5,6]
A = [1,2],B = [3,4,5,6]
A = [1,2,3],B = [4,5,6]
A = [1,2,3,4],B = [5,6]
A = [1,2,3,4,5],B = [6]
A = [1,2,3,4,5,6,B = []
```

Putem rupe o listă în două în punctul în care un element are o anumită valoare. Pentru aceasta apelăm predicatul "append" astfel:

```
append(X,[Element|Y],Z),
```

unde cunoaştem lista Z şi elementul "Element", iar predicatul va returna listele X şi Y. De exemplu, dacă

Z = [1,2,3,4,5,6] şi Element=4

va rezulta:

X = [1,2,3] Y = [5,6].

Două elemente consecutive dintr-o listă pot fi obţinute dacă apelăm predicatul "append" astfel:

```
append(_,[El1,El2|_],Lista).
```

Dacă cunoaştem şi elementul "El1", predicatul va returna valoarea lui "El2", adică valoarea elementului aflat imediat după "El1".

Dacă vrem să verificăm faptul că un element este conţinut într-o listă L cerem:

```
append(_,[Element|_],L).
```

Putem obţine o listă care să conţină ultimul element dintr-o listă dată:

```
append(_,[Ultimul],Lista).
```

Să vedem cum putem folosi predicatul "append" pentru a defini alte predicate. Predicatul "sublist" confirmă faptul că o listă conţine o sublistă. De asemenea acest predicat poate extrage una sau toate sublistele dintr-o listă. Să definim acest predicat:

```
PREDICATES
sublist(REALLIST,REALLIST)
CLAUSES
sublist([],_).
sublist([Head|Tail],List):-
append(_,[Head,Endlist],List),
append(Tail,_,Endlist).
```

Un alt predicat care foloseşte predicatul "append" este cel care permite inversarea ordinii elementelor dintr-o listă.

```
PREDICATES
reverse(INTEGERLIST,INTEGERLIST)
CLAUSES
reverse([],[]).
reverse([H|T],L):
          -reverse(T,Z),append(Z,[H],L).
```

Încercaţi :

```
GOAL Lista = [1,2,3,4,5],
reverse(Lista,Inversa),
  write(Inversa).
```

Să scriem şi alte predicate care permit prelucrarea listelor.

Predicatul următor este util în situaţia în care dorim să aflăm numărul de elemente ale unei liste:

```
PREDICATES
lungime(INTEGERLIST,INTEGER)
CLAUSES
lungime([],0).
lungime([_|Tail],Len):
    -lungime(Tail,Tlen),Len = Tlen + 1.
```

Următorul predicat calculează suma elementelor unei liste de întregi sau de numere reale (după cum se fac declaraţiile în secţiunea PREDICATES).

```
PREDICATES
suma(INTEGERLIST,INTEGER)
CLAUSES
suma([],0):-!.
suma([Num|T],Sum):
        -suma(T,S1),Sum = Num + S1.
```

Dacă în ultima linie se înlocuieşte semnul "+" cu "*" se obţine produsul elementelor.

Predicatul "**sterg**" îndepărtează primul element al unei liste.

```
sterg(X,[X|Z],Z):-!.
sterg(X,[B|L],[B|M]):-sterg(A,L,M).
```

Predicatul "**member**" confirmă sau infirmă apartenenţa unui element la o listă.

```
PREDICATES
member(INTEGER,INTEGERLIST)
CLAUSES
member(X,[X|_]).
member(X,[_|T]):-member(X,T).
```

Şi acest predicat va fi folosit în continuare pentru a defini alte predicate.Următorul predicat, care îl foloseşte şi pe cel anterior definit, ne va permite ştergerea duplicatelor dintr-o listă:

```
PREDICATES
remdup(INTEGERLIST,
        INTEGERLIST)
dupacc(INTEGERLIST,
        INTEGERLIST,INTEGERLIST)
member(INTEGER,INTEGERLIST)
CLAUSES
remdup(L,M):-dupacc(L,[],M).
dupacc([],A,A).
dupacc([H|T],A,L):-member(H,A),
        dupacc(T,A,L).
dupacc([H|T],A,L):-dupacc(T,[H|A],L).
member(X,[X,_]).
member(X,[_|T]):-member(X,T).
GOAL Lista = [1,1,2,2,3,3,4,4,5,6,7],
remdup(Lista,L1),write(L1).
```

Un efect secundar al predicatului "**remdup**" este deteriorarea ordinii iniţiale a elementelor listei după îndepărtarea duplicatelor.O variantă a acestui predicat permite îndepărtarea elementului X dintr-o listă, în situaţia în care acesta este conţinut în mai multe poziţii .

```
PREDICATES
sterg(INTEGER,INTEGERLIST,
INTEGERLIST)
CLAUSES
sterg(_,[],[]).
sterg(X,[X|L],M):-!,sterg(X,L,M).
sterg(X,[Y|L1],Y|L2]):-sterg(X,L1,L2).
```

Cu cîteva modificări, se poate scrie un predicat care înlocuieşte toate elementele X dintr-o listă L cu elementul A, rezultînd lista M:

```
PREDICATES
subst(INTEGER,INTEGERLIST,
        INTEGER,INTEGERLIST)
CLAUSES
subst(_,[],_,[]).
subst(X,[X|L],A,[A|M]):-!,
        subst(X,L,A,M).
subst(X,[Y|L],A,[Y|M]):-
        subst(X,L,A,M).
```

Pentru a rupe o listă în două, prima cu N elemente şi a doua cu restul de elemente, vom defini predicatul "**split**":

```
PREDICATES
split(INTEGER,INTEGERLIST,
        INTEGERLIST,INTEGERLIST)
CLAUSES
split(0,L,[],L):-!.
split(N,[H|L],[H|L1],L2):-
N1 = N-1,split(N1,L,L1,L2).
```

De exemplu:

```
GOAL
Lista = [1,2,3,4,5,6],
split(3,Lista,X,Y),
write(X),nl,write(Y).
```

va returna [1,2,3] şi [4,5,6].

Dacă numărul de elemente după care dorim să rupem lista este mai mare decît numărul de componente, predicatul returnează "fail".

Să definim un predicat care aranjează elementele unei liste în ordine ascendentă sau descendentă, adică sortează o listă:

```
PREDICATES
sort(INTEGERLIST,INTEGERLIST)
sort1(INTEGER,INTEGERLIST,
        INTEGERLIST)
CLAUSES
sort([],[]).
sort([X|L],M):-sort(L,N),
        sort1(X,N,M).
sort1(X,[A|L],[A|M]):-A <= X,!,
        sort1(X,L,M).
sort1(X,L,[X|L]).
```

Dacă în penultima clauză punem condiţia A>=X, ordinea de sortare va fi descendentă.

Pentru a evalua viteza de sortare a acestui predicat să scriem un program care generează 500 de elemente ale unei liste, în mod aleator, le sortează şi cronometrează durata sortării:

```
code = 60
errorlevel = 0
%stack size = 1000
DOMAINS
INTEGERLIST = INTEGER*
PREDICATES
sort(INTEGERLIST,INTEGERLIST)
sort1(INTEGER,INTEGERLIST,
        INTEGERLIST)
generez(INTEGER,INTEGERLIST)
CLAUSES
sort([],[]).
sort([X|L],M):-sort(L,N),sort1(X,N,M).
sort1(X,[A|L],[A|M]):-A <= X,!,
        sort1(X,L,M).
sort1(X,L,[X|L]).
generez(500,X):-!,time(_,_,S,H),
        sort(X,X1),time(_,_,S1,H1),
        write(X1),Sec = S1-S,
  Sutimi = abs(H1-H),nl,write(Sec,
        " Secunde ",Sut," Sutimi").
generez(Contor,L):-random(1000,H),
        X = [H|L],C1 = Contor + 1,!,
generez(C1,X).
GOAL
makewindow(1,15,15,"",0,0,25,80),
        generez(0,[]).
```

Compilaţi acest program şi lansaţi fişierul executabil. Timpul necesar pentru a ordona lista de 500 de elemente este funcţie de viteza calculatorului dvs. (1-4 secunde).Pentru a folosi acest predicat la ordonarea alfabetică a unei liste de şiruri vom rescrie programul nostru astfel:

```
DOMAINS
STRINGLIST = STRING*
PREDICATES
strsort(STRINGLIST,STRINGLIST)
strsort1(STRING,STRINGLIST,
        STRINGLIST)
CLAUSES
strsort([],[]).
strsort([X|L],M):-strsort(L,N),
        strsort1(X,N,M).
strsort1(X,[A|L],[A|M]):-
frontchar(A,A1,_),char_int(A1,A2),
        frontchar(X,X1,_),
        char_int(X1,X2),
        A2 <= X2,!,strsort1(X,L,M).
strsort1(X,L,[X|L]).
```

Observăm că pentru a sorta şiruri ne folosim de un mic truc, care apelează tot la sortarea de întregi şi anume: convertim în întreg primul caracter al

fiecărui string şi deci sortarea se va face ţinînd cont de succesiunea caracterelor în tabela ASCII. O deficienţă a acestei metode este aceea că sortarea se face doar după prima literă a fiecărui şir care compune lista.

Pentru a genera uşor liste, limbajul Prolog ne pune la dispoziţie predicatul predefinit "**findall**" care, după cum sugerează şi denumirea sa, găseşte toate elementele care îndeplinesc o anumită condiţie, din baza de date internă şi generează o listă a acestora, în ordinea în care baza de date există în memorie.

Sintaxa predicatului este următoarea:

findall(C,baza_de_date(A,B,C),
Lista_de_C),

unde C va fi în mod obligatoriu o variabilă liberă, care are rolul de a indica predicatului după care cîmp din baza de date se va genera lista, iar A şi B variabile legate care pun condiţiile de generare a listei "Lista_de_C". Dacă dorim toate elementele din baza de date (fără nici o restricţie), apelul predicatului se va face astfel:

findall(C,baza_de_date(_,_,C),Lista_de_C),

unde "_" desemnează variabila anonimă.Să susţinem afirmaţiile de mai sus cu un program care calculează media de vîrstă a angajaţilor unei întreprinderi:

```
DOMAINS
NUME, ADRESA = STRING
VIRSTA = INTEGER
LISTA = VIRSTA*
DATABASE
evidenta(NUME,ADRESA,VIRSTA)
PREDICATES
sumlist(LISTA,VIRSTA,INTEGER)
CLAUSES
sumlist([],0,0).
sumlist([H|T],Suma,Nr):-
        sumlist(T,S1,N1),
        Suma=H+S1,Nr=N1+1.
GOAL
assertz(evidenta("Vasile Damatale",
        "Str.Republicii Nr 4",35)),
assertz(evidenta("Petre Popescu",
        "Bd. 1848 Nr 122",75)),
assertz(evidenta("Vasilica Tastaman",
        "Str. Lunga Nr7",16)),
findall(C,evidenta(_,_,C),L),
        sumlist(L,S,N),
Media_de_virsta=S/N,
    write("Media= ",Media_de_virsta).
```

Pentru a înţelege mecanismul de funcţionare al predicatelor pe care le-am prezentat, este necesar să ne reamintim unele concepte specifice limbajului Prolog. Cînd apelăm un predicat, care are mai multe argumente, argumentele legate vor fi folosite în clauze pentru a genera argumentele libere, pe care predicatul le va returna. Putem să alegem argumentele de retur (libere), astfel încît un predicat să aibă mai multe întrebuinţări (vezi cazul "append"). Acest proces este denumit în literatura anglo-saxonă "**multiple flow pattern**", adică în traducere (foarte) liberă "model de intrare-ieşire multiplu". Anumite predicate nu acceptă toate variantele de intrare-ieşire. De exemplu "gotowindow(X)" nu poate returna numărul X (La apelarea acestui predicat variabila X va trebui să fie în mod obligatoriu legată de o valoare întreagă). În caz contrar compilatorul va genera mesajul de eroare "*This flow pattern does not exist.(Look in the message window)*" şi ne va arăta în fereastra de mesaje varianta I/O greşită. Alte predicate acceptă variante multiple de I/O, ca de exemplu predicatul "shiftwindow(X)", care poate fi atît de intrare cît şi de ieşire. Aşadar, în funcţie de starea variabilelor (libere sau legate) comportarea predicatelor va fi diferită.

Un alt aspect important este cel legat de proprietatea predicatelor de a se apela pe ele însele. Această proprietate se numeşte recursivitate. Recursivitatea este SINGURA metodă pe care limbajul Prolog ne-o pune la dispoziţie pentru a crea bucle. Să analizăm fenomenul de recursivitate în cazul predicatului "member":

```
member(X,[X|_]).
member(X,[_,T]):-member(X,T).
```

Prima clauză spune: "Un element X este menbru al unei liste, dacă este conţinut în capul listei". Primul argument X este legat de valoarea pe care dorim să o găsim încă de la apelarea predicatului. Al doilea argument, lista de explorat, este desfăcut în "cap", a cărui valoare încă nu o ştim, şi coada listei, care nu ne interesează (variabial anonimă "_").Aşadar, se testează egalitatea dintre primul X (care este legat) şi al doilea X, care este primul element al listei. Dacă egalitatea nu este îndeplinită, trecem la a doua clauză care spune: "Dacă nu, elementul X este membru al listei dacă este conţinut în coada sa şi îl vom căuta acolo apelînd din nou prima clauză, pentru lista din care am îndepărtat primul element!" Astfel, prin recursivitate se caută elementul din capul listei prin îndepărtarea succesivă a cîte unui element şi explorarea listei rămase.

În orice limbaj de programare recursivitatea depune în memoria "stivă" parametrii de dinaintea apelului. Acest lucru, în Prolog, poate fi atît un avantaj cît şi un dezavantaj.

Să punem răul în faţă şi să analizăm dezavantajul. Este evident că în anumite situaţii riscăm să umplem stiva înainte ca treaba noastră să fie gata. În acest caz Prologul va genera mesajul de eroare "*Stack overflow. Reconfigure with Options if necessary*". în această situaţie putem mări zona de memorie alocată stivei (în detrimentul memoriei de uz general "heap") prin mărirea numărului de paragrafe din "Options/Compiler directives/Memory allocation/Stack size", opţiune accesibilă înaintea generării formatului executabil. Dacă programul este conceput eronat, acesta va consuma rapid stiva, oricît de mare este aceasta. Prin plasarea predicatului "!" (cut) în locurile corespunzătoare, acesta va indica programului că nu este nevoie să reţină în stivă parametrii de dinaintea sa şi deci putem să facem economie substanţială de memorie şi viteză. În schimb dacă abuzăm de serviciile acestui predicat pierdem toate avantajele mecanismului de revenire (backtrack), care este esenţa limbajului.

Cu afirmaţia de mai sus am atins aspectul nr.2, şi anume avantajele memorării parametrilor generaţi de recursivitate. Singura modalitate a limbajului de a reveni şi a reevalua situaţiile în care nu poate îndeplini scopul propus este apelul la stivă, pentru a relua lucrul de la situaţia anterioară, în scopul abordării altei variante. Acest mecanism este folosit masiv în programul de sortare şi algoritmul său nu poate fi folosit în alte limbaje tocmai datorită absenţei acestui mecanism. Într-o formulare foarte aproximativă putem spune că algoritmul de sortare forţează rearanjarea listei de prelucrat pînă cînd este îndeplinită condiţia ca toate elementele Xi să fie mai mici decît Xi+1.

(continuare în pag. 55)

LECOM mita

Distribuitori autorizati LECOM - MITA

- 15.04.92 -

Bacau SAITA S.R.L.
Str. Cremenea nr. 2, apt. 5
Tel.: *931* / 49544

SARMIS S.R.L.
Str. Bucegi nr. 115, sc. B, apt. 83
Tel.: *931* / 57443

Baia Mare SINTEC S.R.L.
Str. Culturii nr. 5, C.P. 638
Tel.: *994* / 13067 15720/171
Fax: *994* / 13067

Braila PROCONS S.R.L.
Str. Viziru nr. 3, C.P. 1009 O.P. 10
Tel.: *946* / 14306
Fax: *946* / 34091

Brasov ELCATE S.R.L.
Str. Carpenului nr. 10/6
Tel.: *921* / 16720 / 1403

Bucuresti BUGAS COMPUTERS S.R.L.
Calea Calarasi nr.23, bl.105B, apt.25
Tel.: *90* / 129295 152346 155072
Fax: *90* / 129295 155072

INTER AS Ltd. S.R.L.
Bd. Magheru nr. 9, sect. 1
Tel.: *90* / 159254

LAR S.A.
Str. Stirbei Voda nr. 2-4
Tel.: *90* / 153276

Buzau GEMI S.R.L.
Str. Pompiliu Stefan nr. 6A
Tel.: *974* / 12323

Cluj-Napoca SCAAN S.R.L.
Str. Constanta nr. 10
Tel.: *95* / 134463

Constanta INTERNATIONAL MARKETING COMPANY S.R.L.
Str. Stefan cel Mare,
Magazinul Universal "Tomis" - parter
Tel.: *91* / 641750 643079
Fax: *91* / 641750

SELCOM S.R.L.
Aleea Murelor nr. 8, bl. BC6, apt. 60
Tel.: *91* / 614630

Craiova HELCO S.R.L.
Str. N. Balcescu nr. 49
Tel.: *94* / 116800
Fax: *94* / 133401

Dej GENERAL TELECOM S.R.L.
Str. Mihail Kogalniceanu nr. 2
Tel.: *95* / 214738
Fax: *95* / 214738

Deva ELECTROMETAL S.R.L.
Str. Iosif Vulcan nr. 44
Tel.: *956* / 22388

Iasi INTERELECTRIC S.A.
Str. Elena Doamna nr. 24
Tel.: *981* / 12447
Fax: *981* / 19834

Miercurea-Ciuc K ELECTRIC & ELECTRONIC S.R.L.
Str. Iancu de Hunedoara nr. 29
Tel.: *958* / 16797
Fax: *958* / 16797

Oradea LIDER IMPEX S.R.L.
Str. Transilvaniei nr. 5,
Magazinul Tineretului
Tel.: *991* / 12757 63457
Fax: *991* / 12757

Petrosani INFOTEL PUB SERVICE S.R.L.
Str. Independentei, bl. 18, apt. 17
Tel.: *935* / 42450

Piatra-Neamt TRAPEL S.N.C.
Str. Viforului, bl. C1, apt. 2
Tel.: *936* / 22121

Pitesti WORTH S.R.L.
Str. Maior Sontu, bl. D1, apt. 21
Tel.: *976* / 24054
Fax: *976* / 24054

Ploiesti EMSERV S.R.L.
Str. Clementei nr. 41
Tel.: *971* / 44338 63054

SEPRAS ELECTRONIC S.R.L.
Str. Frasinet nr. 9
Tel.: *971* / 52572

Rimnicu-Vilcea GALAX S.R.L.
Str. Libertatii, bl. 3, apt. 2
Tel.: *947* / 21392

Satu-Mare DATA ELECTRONIC S.R.L.
Cap Pod Decebal, bl. C23/14
Tel.: *997* / 12261 37131

Sibiu PRAXIS COMPUTER S.R.L.
Aleea Buia nr. 2, apt. 23
Tel.: *924* / 18101

Suceava DATA SERVICE S.N.C.
Str. George Enescu nr. 16
Tel.: *987* / 16747 / 115 26248 / 115

Timisoara MANUEL S.R.L.
Str. Iosif Vulcan nr. 15
Tel.: *96* / 138533

XERICO S.R.L.
Str. G. Cosbuc nr. 5, sc. B, apt. 1
Tel.: *96* / 133586

Tirgu-Mures MICRO ATCI S.R.L.
Str. Gh. Doja nr. 36
Tel.: *954* / 31660
Fax: *954* / 31660

Turnu-Severin AGENTIA PROMOVARE TEHNICA SI INDUSTRIALA
Str. Mihai Eminescu nr. 38
Tel.: *978* / 14250

LECOM-SRL • Calea Victoriei nr. 126 • 70179 BUCURESTI • ROMÂNIA • Tel.: 50.64.20; 50.79.94; 12.95.58 • Fax: 12.95.58 • Tlx.: 10807
BRCE: acc. no. 300.8.363-7 (lei) • MRB: acc. no. 123/90 (US$) • Reg. no. 40 / 441 / 15.02.91 • Registered capital 25000 US$

Curs de programare OOP în C++, partea a treia

Dinamica şi construcţia claselor

La programe mai mari, gestionarea grijulie a spaţiului de memorie este tot mai necesară. În ciuda Mega- şi Gigaocteţilor ce devin disponibili, programarea bună (adică eficientă în ce priveşte gestiunea memoriei) ar trebui să ţină de "bon-ton". Oricum, posibilităţi de administrare dinamică a memoriei există de mult, deci este normal ca aceste concepte să fi fost preluate şi în C++.

Pînă acum, în C, se lucra cu "alloc( )", "malloc( )" şi "stackavail( )". Aceste funcţii erau preluate prin intermediul fişierului-header "malloc.h" respectiv "alloc.h". Acum, aceste posibilităţi sînt oferite, mult mai elegant, de-a dreptul din limbaj.

Nu mai este necesar calculul de memorie necesară pentru un obiect, pentru a vedea apoi dacă acest necesar este disponibil. Acest lucru este preluat de compilator.

Mulţi utilizatori au oroare sau se tem de lucrul cu obiecte dinamice. Gestionarea lor ar fi prea complicată, erori mici ar duce la blocarea sistemului. Cît de simplu este, de fapt, se poate vedea în listingul EX74.CPP. Acesta are modificări minore faţă de EX73.CP (prezentat în episodul trecut). N-au fost operate decît adaptările pentru obiecte dinamice. Clasele şi metodele au fost preluate fără modificări. Numai pentru instanţierile individuale ale claselor utilizate au trebuit declaraţi pointeri, spre clase, prin intermediul lui "new". Aceste instanţieri sînt şterse, după utilizarea în partea corespunzătoare de program, prin apelul lui "delete".

Înainte de analiza mai profundă a obiectelor dinamice, o scurtă incursiune în "constructori". Spre deosebire de dinamică, care în exemplele precedente nici n-a fost utilizată, constructorii unei clase au fost folosiţi deseori. Fiecare clasă, în momentul creeri ei (adică atunci cînd i se alocă pentru prima dată memorie) poate apela la constructori. Cu ajutorul lor, unele atribute ale unei clase pot fi iniţializate la creere. Ceea ce s-a şi făcut.

Opusul constructorilor îl reprezintă destructorii. Ei sînt secvenţe de instrucţiuni ce trebuiesc executate la "desfiinţarea" unui obiect. Pînă acum, toate exemplele date au lucrat exclusiv cu obiecte statice, astfel încît nu a fost necesară definirea unor destructori suplimentari. Pentru cazul lipsei constructorilor sau destructorilor, se execută rutine standard.

```
1   // Exemplul 7.4
2   // obiecte dinamice
3
4   #include <conio.h>
5   #include <stdio.h>
6   #include <stdlib.h>
7
8   const XMax = 80;
9   const YMax = 24;
10  const XMin = 1;
11  const YMin = 1;
12  enum Boolean {false, true};
13
14  void Expand (char w, char a, int l, char *t)
15  {
16    int i;
17
18    for (i=0; i<=2*l; i=i+2) {
19      t[i] = w; t[i+1] = a;
20    }
21  }
22
23  class Semn  // defineste clasa Semn
24  {
25    int x, y; // coordonatele
26
27    public:
28    char valoare;    // valoare alfanumerica
29    char attr; // atribut ecran
30    Semn::Semn (int InitX, int InitY)
31    {
32      x = InitX;
33      y = InitY;
34    }
35    int Semn::GetX () {return x;}
36    int Semn::GetY () {return y;}
37    void Semn::Position (int PosX, int PosY)
38    {
39      x = PosX; y = PosY;
40    }
41    void Semn::Citire()
42    {
43      gettext (x, y, x, y, &valoare);
44    }
45    virtual void Semn::Afisare ()
46    {
47      puttext (x, y, x, y, &valoare);
48    }
49    void Semn::Scriere (char Sign)
50    {
51      valoare = Sign;
52      Afisare ();
53    }
54    void Semn::BackColor ()
55    {
56      int Color;
57
58      if (attr > 128) {
59        Color = attr - 128;
60        Color = _rotr (Color, 4);
61        attr = Color + 16 * Color + BLINK;
62      }
63      else {
64        attr = _rotr (attr, 4);
65        attr = attr + 16 * attr;
66      }
67    }
68    void Semn::Stergere ()
69    {
70      char Mvaloare, Mattr;
71      // dupa stergere, in obiect ramin nemodificate
72      // datele pentru valoare si atribut !!!
73
74      Mvaloare = valoare; Mattr = attr;
75      BackColor ();
76      valoare = ' ';
77      Afisare ();
78      valoare = Mvaloare; attr = Mattr;
79    }
80    void Semn::Miscare (int NewX, int NewY)
81    {
82      Stergere ();
83      Position (NewX, NewY);
84      Afisare ();
85    }
86    void Semn::Deplasare (int difx, int dify)
87    {
88      Stergere ();
89      Position (x+difx, y+dify);
90      Afisare ();
91    }
92    void Semn::Copiere (int NewX, int NewY)
93    {
94      Position (NewX, NewY);
95      Afisare ();
96    }
97    void Semn::Set_attr (char fattr, char battr = 0,\
98          char Blink = 0)
99    {
100     if (battr < 8)
101       attr = fattr + 16 * battr + Blink;
102     else
103       attr = fattr + Blink;
104   }
105 };
106
107 class Linie : public Semn
108 {
109   protected:
110   int LungX;
111   public:
112   Linie::Linie (int InitX, int InitY)\
113       : Semn (InitX, InitY)
114   {
115     LungX = 0;
116   }
117   int Linie::GetL () { return LungX; }
118   int Linie::Stinga ()
119   {
120     int x1;
121     x1 = GetX() - XMin + 1;
122     if (GetX() < XMin)
123       {
124         if (GetX()+LungX > XMin)
125           x1 = XMin;
126         else
127           x1 = 0;
128       }
129     return (x1);
130   }
131   int Linie::Dreapta ()
132   {
133     int x2;
134     x2 = GetX() + LungX - XMin;
135     if (GetX()+LungX > XMax)
136       {
137       if (GetX() <= XMax)
138         x2 = XMax;
```

```
      else
        x2 = 0;
    }
    return (x2);
}
void Linie::Afisare ()
{
    char temp[4096];
    int x1, x2;

    x1 = Stinga();
    x2 = Dreapta();
    Expand (valoare, attr, x2-x1, temp);
    if (x1 != 0)
      if (x2 != 0)
        puttext (x1, GetY(), x2, GetY(), &temp);
}
void Linie::Fix_lungime (int L)
{
    LungX = L;
}
void Linie::Dilatare ()
{
    Stergere ();
    LungX += 1;
    Afisare ();
}
void Linie::Deformare ()
{
    Stergere ();
    if (LungX > 0) LungX -= 1;
    Afisare ();
}
};

class Dreptunghi : public Linie
{
    int LungY;

public:
Dreptunghi::Dreptunghi (int InitX, int InitY)\
        : Linie (InitX, InitY)
{
    LungX = 0;
    LungY = 0;
}
void Dreptunghi::Afisare ()
{
    char temp [4096];
    int x1, x2;

    x1 = Stinga();
    x2 = Dreapta();
    Expand (valoare, attr, (x2-x1+1)*LungY, temp);
    if (x1 != 0)
      if (x2 != 0)
        puttext (x1, GetY(),\
              x2, GetY()+LungY-YMin, &temp);
}
void Dreptunghi::Fix_lungime (int lx, int ly)
{
    LungX = lx;
    LungY = ly;
}
void Dreptunghi::Dilatare (int x1, int y1)
{
    Stergere ();
    LungX += x1;
    LungY += y1;
    Afisare ();
}
void Dreptunghi::Deformare (int x1, int y1)
{
    Stergere ();
    if (LungX > x1) LungX = LungX - x1;
    if (LungY > y1) LungY = LungY - y1;
    Afisare();
}
};

int main()
{
    int x1, y1, s, z;
    char tasta;
    Boolean sfirsit;
    // fond cu semn albastru
    x1 = 1; y1 = 1;
    Semn *BSemn = new Semn (x1, y1);
    BSemn->Scriere (219);
    BSemn->Set_attr (BLUE);
    for (s=0; s<25; s++)
      for (z=0; z<80; z++)
        BSemn->Copiere (x1+z, y1+s);
    delete BSemn;
    printf ("Apasati o tasta ...");
    getch ();
    // Reprezinta si apoi modifica o linie
    x1 = 10; y1 = 10;
    Linie *LSemn = new Linie (x1, y1);
    LSemn->Set_attr (RED, BLUE);
    LSemn->Fix_lungime (60);
    LSemn->Scriere ('-');
    do
    {
      sfirsit = true;
      tasta = getch ();
      if (tasta == 0)
        {
          sfirsit = false;
          tasta = getch();
          switch (tasta)
            {
              case 72: LSemn->Deplasare (0, -1);
                       break;
              case 80: LSemn->Deplasare (0, 1);
                       break;
              case 75: LSemn->Deformare ( );
                       break;
              case 77: LSemn->Dilatare ( );
                       break;
              // altceva decit tasta cursor
              default: sfirsit = true;
            };
        };
    } while (!sfirsit);
    LSemn->Stergere ();
    delete LSemn;
    getch ();
    // Reprezinta si apoi modifica un dreptunghi
    x1 = 10; y1 = 2;
    Dreptunghi *DSemn = new Dreptunghi (x1, y1);
    DSemn->Set_attr (CYAN, BLUE);
    DSemn->Fix_lungime (15, 10);
    DSemn->Scriere (219);
    do
    {
      sfirsit = true;
      tasta = getch ();
      if (tasta == 0)
        {
          sfirsit = false;
          tasta = getch ();
          switch (tasta)
            {
              case 72:  DSemn->Deplasare (0, -1);
                        break;
              case 80: DSemn->Deplasare (0, 1);
                        break;
              case 75: DSemn->Deplasare (-1, 0);
                        break;
              case 77: DSemn->Deplasare (1, 0);
                        break;
              // nu e tasta cursor
              default: sfirsit = true;
            };
        };
    } while (!sfirsit);
    DSemn->Scriere ('=');
    do
    {
      sfirsit = true;
      tasta = getch ();
      if (tasta == 0)
        {
          sfirsit = false;
          tasta = getch ();
          switch (tasta)
            {
              case 72: DSemn->Deformare (0, 1);
                       break;
              case 80: DSemn->Dilatare (0, 1);
                       break;
              case 75: DSemn->Deformare (1, 0);
                       break;
              case 77: DSemn->Dilatare (1, 0);
                       break;
              // nu e tasta cursor
              default: sfirsit = true;
            };
        };
    } while (!sfirsit);
    DSemn->Stergere ();
    delete DSemn;
    getch ();
    clrscr ();
    return 0;
}
```

EX74.CPP

La clase derivate succesiv, apare problema succesiunii apelurilor constructorilor efectuate de compilator. Dacă clasa de bază a fost definită ca şi clasă virtuală, atunci constructorul clasei de bază va fi executat înainte de constructorii "derivaţi". Altfel, programul apelează constructorii diverselor clase în ordinea declarării lor. Ceea ce pentru clasa "Dreptunghi", derivată din clasa "Linie", înseamnă că întîi intră în joc constructorul din "Linie", apoi cel din "Dreptunghi". De exemplu la

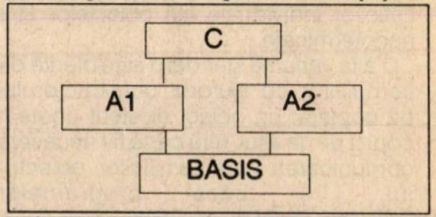
class Dreptunghi: public Punct, public Linie

este apelat întîi constructorul "Punct", apoi cel din "Linie" şi apoi cel din "Dreptunghi".

În cazul claselor de bază virtuale, lucrurile se desfăşoară altfel.

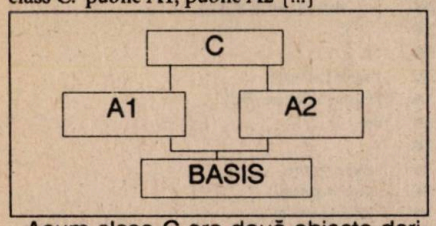
Nu este posibilă moştenirea directă a unei clase de bază spre o clasă derivată

class C: public BAZA, public BAZA {...}.



Acest lucru se poate face numai indirect:

```
class A1:    public BASIS {...}
class A2:    public BASIS {...}
class C: public A1, public A2 {...}
```



Acum clasa C are două obiecte derivate ale clasei BASIS. Aceasta poate genera probleme relativ la univocitatea sensului; acestea sînt ocolite prin inserarea cuvîntului cheie "virtual". Atunci şi clasă de bază "BASIS" se mai numeşte "clasă de bază virtuală". Lucrurile se mai pot reprezenta şi formula în felul următor:

```
class A1:    virtual public BASIS {...}
class A2:    virtual public BASIS {...}
class C:     public A1, public A2 {...}
```

Un obiect al clasei "C" are atunci un singur subobiect al clasei de bază BASIS. Nu există însă nici o relaţie între clasele derivate A1 şi A2.

După această scurtă incursiune în clasele de bază virtuale, să ne întoarcem la succesiunea apelurilor constructorilor. Cei ai claselor de bază virtuale vor fi apelaţi de constructorii claselor nevirtuale. Dacă există mai multe clase de bază virtuale, atunci se consideră ordinea declarării lor. Dar prin ce se recunoaşte un constructor?

```
class X
    {    int y;
         public
         X (int z=0) {y=z}//constructor
    };
[classname::]classname{...}
```

Parantezele drepte indică faptul că aceste simboluri pot fi omise, dacă constructorul este descris în interiorul clasei.

Pentru destructor, se foloseşte caracterul "tilda" (" ~ ").

```
[classname::]~classname {...}
```

În general se poate spune: fiecare clasă dispune de obicei de o metodă (funcţie element) cu propriul ei nume, care conţine construcţia clasei. Analog, relativ la "destrucţie".

Chiar dacă nu se indică un constructor, compilatorul generează unul, aşanumitul constructor standard. La apelul acestuia "X::X()" nu se predau parametri, astfel încît valorile atribuite caracteristicilor individuale ale obiectelor sînt nedeterminate.

O altă variantă standard este oferită de compilator cu ajutorul constructorului de copiere: un obiect existent poate fi copiat pe un altul, fără ca să fie necesară comunicarea caracteristicilor obiectului. Pe baza programelor STDCNSTR.CPP şi CPYCNSTR.CPP, această stare de fapt poate fi înţeleasă mai bine. De altfel, în figura 2, pentru obiectele "doi" şi "trei", constructorul de copiere este utilizat "din plin". În final, toate trei obiectele au aceeaşi valoare (aleatoare) în "y". În programul CPYCNSTR.CPP, numai primul obiect are valoare nulă, deoarece nu a fost transmis nici un parametru. Toate celelalte 3 obiecte, cu numele "doi", "trei" şi "patru" au valoarea 10, deoarece ele au fost create prin copiere.

```
1   //Stdcnstr.cpp
2   // Constructor standard
3
4   #include <iostream.h>
5
6   class X
7   {
8    int y;
9   public:
10   GetY()   { return y; }
11  };
12
13  main()
14  {
15   X unu;
16   cout << "Obiectul unu: ";
17   cout << unu.GetY () << "\n";
18
19   X doi = unu;
20   cout << "Obiectul doi: ";
21   cout << doi.GetY () << "\n";
22
23   X trei (doi);
24   cout << "Obiectul trei: ";
25   cout << trei.GetY () << "\n";
26  }
27
```

STDCNSTR.CPP

```
1   // Cpycnstr.cpp
2   // Copy Constructor
3
4   #include <iostream.h>
5
6   class X
7   {
8    int y;
9   public:
10   X (int z = 0) { y = z; }
11   GetY()   { return y; }
12  };
13
14  main()
15  {
16   X unu;
17   cout << "Obiectul unu: ";
18   cout << unu.GetY () << "\n";
19
20   X doi(10);
21   cout << "Obiectul doi: ";
22   cout << doi.GetY () << "\n";
23
24   X trei = doi;
25   cout << "Obiectul trei: ";
26   cout << trei.GetY () << "\n";
27
28   X patru(trei);
29   cout << "Obiectul patru: ";
30   cout << patru.GetY() << "\n";
31  }
32
```

CPYCNSTR.CPP

Stream-uri în C++

Constructorii şi destructorii nu returnează vreo valoare sau un tip. Spre deosebire de constructori, destructorii nu au niciodată parametri. La variabile globale, destructorii sînt apelaţi la sfîrşitul funcţiei "main ( )". Succesiunea de apel a destructorilor este inversă celei a constructorilor.

La clase derivate, destructorul clasei de bază poate fi declarat ca virtual. După care şi cei ai claselor derivate sînt virtuali. Prin aceasta se asigură parcurgerea tuturor destructorilor ierarhiei claselor.

În listingul CODP1.CPP se realizează legătura celor două teme majore ale părţii a treia a cursului: se utilizează, pe lîngă constructori şi destructori, şi administrarea dinamică a memoriei. În cadrul constructorilor, spaţiul de memorie necesar este cerut prin "new". Acest spaţiu de memorie este eliberat, după execuţia programului, de către destructorul corespunzător.

În exemplul dat apare o altă caracteristică fundamentală a C++- ului: utilizarea streamurilor (în engleză, "stream"=fluviu).

Ca cele mai multe lucruri, stream-urile din versiunea 1.2 respectiv forma lor extinsă din 2.0 îşi au originea în lexicul de bază al C-ului. Ele provin din unităţile standard "stdin", "stdout" şi "stderr".

Cu ajutorul stream-urilor se tratează cele mai importante sarcini care apar în cursul operaţiilor de intrare/ieşire. Acestea cuprind şi intrări-ieşiri de fişiere sau şiruri. La ieşiri, se ţine seama de conversii numerice sau în şiruri de caractere şi de formatări generale. La intrări se face o prelucrare a aracterelor "whitespace" ('u',\t,\r,\n,\f,\v). Aceste caractere au în comun faptul că la apelul funcţiei "isspace ( )" returnează aceeaşi valoare.

Programele C++ "startează" cu trei streamuri predefinite deschise. Acestea se numesc "cin", "cout" şi "cerr" şi corespund intrării standard, ieşirii standard şi ieşirii standard în caz de eroare din C (stdin, stdout şi stderr).

Stream-ul de ieşire a fost legat de operatorul "<", care a fost supraîncărcat. El leagă partea stîngă şi partea dreaptă, pentru a returna operandul stîng. De exemplu,

```
cout < "Hello world\n";
```

are ca efect scrierea şirului "Hello world" înspre "cout". Totodată, "cout" preia şi sarcinile de formatare, rezolvate asemănător ca în printf. Dacă "i" este de tip "int" iar "l" de tip "long", atunci instrucţiunile

```
cout < i < " " < l;
printf ("%d %l", i, l);
```

```
1   // codp1.cpp
2   // constructori si destructori
3   // clase dinamice
4
5   #include <stream.h>
6
7   class codp {
8    char *nr;
9    char *localitate;
10  public:
11   codp();
12    ~codp();
13   void introd (ostream);
14  };
15
16  codp::codp()
17  {
18   nr = new char [6];
19   localitate = new char [30];
20  }
21
22  codp:: ~codp()
23  {
24   delete[6] nr;
25   delete[30] localitate;
26  }
27
28  void codp::introd(ostream out)
29  {
30   cout << "codp: ";
31   cin >> nr;
32   out << nr;
33
34   cout << "Localitatea: ";
35   cin >> localitate;
36   out << localitate << "\n";
37  }
38
39  main()
40  {
41   codp c1;   // instantiere codp
42   filebuf f;  // tampon pentru stream-ul de iesire
43
44   if (f.open ("codp.dat", append) == 0)
45    cerr << "Fisierul de coduri postale \
46          nu poate fi deschis.  \n";
47   else {
48    ostream strim(&f);
49    c1.introd (strim);
50   }
51  }
```

CODP1.CPP

au exact acelaşi efect. Se poate aprecia care din cele două modalităţi de exprimare este mai clară.

Deoarece în exemplul CODP1.CPP se folosesc stream-urile versiunii 1.2, la utilizarea Borland C++ 2.0 aceste clase şi metode trebuiesc adăugate la linkeditare din biblioteca "OLDSTRMx".

Accesul la stream-urile vechi se recunoaşte după instrucţiunea de includere a STREAM.H, care încă nu apelează IOSTREAM.H. Acest lucru se întîmplă abia în fişierul FSTREAM.H, astfel încît acolo e utilizabilă toată panoplia stream-urilor din 2.0. Sintaxa generală, simplificată a linkeditorului este

tlink <obj>, <exe>, <map>, <lib>

Înainte de prima virgulă se specifică toate fişierele obiect ce urmează să fie linkeditate. După ultima virgulă sînt specificate toate bibliotecile ce trebuiesc utilizate pentru ca să nu apară erori de genul "undefined..." în timpul linkeditării.

tlink cox, codp1, codp2, , oldstrmx cx

Funcţie de modelul de memorie se deosebesc cîteva fişiere. "x" caracterizează fişierele dependente de modelul de memorie. În "c0x" este conţinut codul de pornire. Biblioteca funcţiilor runtime este memorată în modulul "cx", în timp ce funcţiile mai vechi relative la stream-uri din versiunea 1 2 se află în "oldstrmx". Apelul concret pentru modelul "small".

tlink cos plz1, plz1,, oldstrms cs

```
1   // codp2.cpp
2   // constructori si destructori
3   // operatori supraincarcati
4
5   #include <fstream.h>
6
7   class codp {
8    char *nr;
9    char *localitate;
10  public:
11   codp();
12    ~codp();
13   void introd ();
14   friend ostream& operator << (ostream&, codp);
15  };
16
17  ostream& operator<<(ostream& out, codp c)
18  {
19   out << c.nr << c.localitate << "\n";
20   return out;
21  }
22
23  codp::codp()
24  {
25   nr = new char [6];
26   localitate = new char [30];
27  }
28
29  codp:: ~codp()
30  {
31   delete[6] nr;
32   delete[30] localitate;
33  }
34
35  void codp::introd()
36  {
37   cout << "codp: ";
38   cin >> nr;
39
40   cout << "Localitatea: ";
41   cin >> localitate;
42  }
43
44  main()
45  {
46   codp c1;   // instantiere codp
47   filebuf f;  // tampon pentru stream-ul de iesire
48
49   if (f.open ("codp2.dat", ios::out + ios::app) == 0)
50    cerr << "Fisierul de coduri postale \
51          nu poate fi deschis.  \n";
52   else {
53    ostream strim(&f);
54    c1.introd ();
55    strim << c1;
56   }
57  }
```

CODP2.CPP

```
1   // Exemplul 7.5
2   // obiecte dinamice, fara constructori
3
4   #include <conio.h>
5   #include <stdio.h>
6   #include <stdlib.h>
7
8   const XMax = 80;
9   const YMax = 24;
10  const XMin = 1;
11  const YMin = 1;
12  enum Boolean {false, true};
13
14  void Expand (char w, char a, int l, char *t)
15  {
16   int i;
17
18   for (i=0; i<=2*l; i=i+2) {
19    t[i] = w; t[i+1] = a;
20   }
21  }
22
23  class Semn  // defineste clasa Semn
24  {
25   int x, y; // coordonatele
26
27   public:
28   char valoare;    // valoare alfanumerica
29   char attr; // atribut ecran
30  // Semn::Semn (int InitX, int InitY)
31  // {
32  //  x = InitX;
33  //  y = InitY;
34  // }
35   // functii-element publice
36   int Semn::GetX () {return x;}
37   int Semn::GetY () {return y;}
38   void Semn::Position (int PosX, int PosY)
39   {
40    x = PosX; y = PosY;
41   }
42   void Semn::Citire()
43   {
44    gettext (x, y, x, y, &valoare);
45   }
46   virtual void Semn::Afisare ()
47   {
48    puttext (x, y, x, y, &valoare);
49   }
50   void Semn::Scriere (char Sign)
51   {
52    valoare = Sign;
53    Afisare ();
54   }
55   void Semn::BackColor ()
56   {
57    int Color;
58
59    if (attr > 128) {
60     Color = attr - 128;
61     Color = _rotr (Color, 4);
62     attr = Color + 16 * Color + BLINK;
63    }
64    else {
65     attr = _rotr (attr, 4);
66     attr = attr + 16 * attr;
67    }
68   }
69   void Semn::Stergere ()
70   {
71    char Mvaloare, Mattr;
72    // dupa stergere, in obiect ramin nemodificate
73    // datele pentru valoare si atribut !!!
74
75    Mvaloare = valoare; Mattr = attr;
```

```
   BackColor ();
   valoare = ' ';
   Afisare ();
   valoare = Mvaloare; attr = Mattr;
  }
  void Semn::Miscare (int NewX, int NewY)
  {
   Stergere ();
   Position (NewX, NewY);
   Afisare ();
  }
  void Semn::Deplasare (int difx, int dify)
  {
   Stergere ();
   Position (x+difx, y+dify);
   Afisare ();
  }
  void Semn::Copiere (int NewX, int NewY)
  {
   Position (NewX, NewY);
   Afisare ();
  }
  void Semn::Set_attr (char fattr, char battr = 0,\
                       char Blink = 0)
  {
   if (battr < 8)
     attr = fattr + 16 * battr + Blink;
   else
     attr = fattr + Blink;
  }
 };

 class Linie : public Semn
 {
  protected:
  int LungX;
  public:
 // Linie::Linie (int InitX, int InitY)\
 //      : Semn (InitX, InitY)
 // {
 //  LungX = 0;
 // }

  int Linie::GetL () { return LungX; }
  int Linie::Stinga ()
  {
   int x1;
   x1 = GetX() - XMin + 1;
   if (GetX() < XMin)
     {
       if (GetX()+LungX > XMin)
         x1 = XMin;
       else
         x1 = 0;
     }
   return (x1);
  }
  int Linie::Dreapta ()
  {
   int x2;
   x2 = GetX() + LungX - XMin;
   if (GetX()+LungX > XMax)
     {
       if (GetX() <= XMax)
         x2 = XMax;
       else
         x2 = 0;
     }
   return (x2);
  }
  void Linie::Afisare ()
  {
   char temp[4096];
   int x1, x2;

   x1 = Stinga();
   x2 = Dreapta();
   Expand (valoare, attr, x2-x1, temp);
   if (x1 != 0)
    if (x2 != 0)
      puttext (x1, GetY(), x2, GetY(), &temp);
  }
  void Linie::Fix_lungime (int L)
  {
   LungX = L;
  }
  void Linie::Dilatare ()
  {
   Stergere ();
   LungX += 1;
   Afisare ();
  }
  void Linie::Deformare ()
  {
   Stergere ();
   if (LungX > 0) LungX -= 1;
   Afisare ();
  }
 };

 class Dreptunghi : public Linie
 {
  int LungY;

  public:
  // Dreptunghi::Dreptunghi (int InitX, int InitY)\
  //         : Linie (InitX, InitY)
  // {
  //  LungX = 0;
  //  LungY = 0;
  // }

  void Dreptunghi::Afisare ()
  {
   char temp [4096];
   int x1, x2;

   x1 = Stinga();
   x2 = Dreapta();
   Expand (valoare, attr, (x2-x1+1)*LungY, temp);
   if (x1 != 0)
    if (x2 != 0)
      puttext (x1, GetY(),\
               x2, GetY()+LungY-YMin, &temp);
  }
  void Dreptunghi::Fix_lungime (int lx, int ly)
  {
   LungX = lx;
   LungY = ly;
  }
  void Dreptunghi::Dilatare (int x1, int y1)
  {
   Stergere ();
   LungX += x1;
   LungY += y1;
   Afisare ();
  }
  void Dreptunghi::Deformare (int x1, int y1)
  {
   Stergere ();
   if (LungX > x1) LungX = LungX - x1;
   if (LungY > y1) LungY = LungY - y1;
   Afisare();
  }
 };

 int main()
 {
  int x1, y1, s, z;
  char tasta;

  Boolean sfirsit;
  // fond cu semn albastru
  x1 = 1; y1 = 1;
  Semn *BSemn = new Semn ();
  BSemn->Position (x1,y1);
  BSemn->Scriere (219);
  BSemn->Set_attr (BLUE);
  for (s=0; s<25; s++)
   for (z=0; z<80; z++)
     BSemn->Copiere (x1+z, y1+s);
  delete BSemn;
  printf ("Apasati o tasta ...");
  getch ();
  // Reprezinta si apoi modifica o linie
  x1 = 10; y1 = 10;
  Linie *LSemn = new Linie ();
  LSemn->Position (x1,y1);
  LSemn->Set_attr (RED, BLUE);
  LSemn->Fix_lungime (60);
  LSemn->Scriere ('-');
  do
  {
   sfirsit = true;
   tasta = getch ();
   if (tasta == 0)
     {
       sfirsit = false;
       tasta = getch();
       switch (tasta)
         {
           case 72: LSemn->Deplasare (0, -1); // sus
                    break;
           case 80: LSemn->Deplasare (0, 1);  // jos
                    break;
           case 75: LSemn->Deformare ();  // stinga
                    break;
           case 77: LSemn->Dilatare ();  // dreapta
                    break;
           // altceva decit tasta cursor
           default: sfirsit = true;
         };
     };
   } while (!sfirsit);
  LSemn->Stergere ();
  delete LSemn;
  getch ();
  // Reprezinta si apoi modifica un dreptunghi
  x1 = 10; y1 = 2;
  Dreptunghi *DSemn = new Dreptunghi ();
  DSemn->Position (x1,y1);
  DSemn->Set_attr (CYAN, BLUE);
  DSemn->Fix_lungime (15, 10);
  DSemn->Scriere (219);
  do
  {
   sfirsit = true;
   tasta = getch ();
   if (tasta == 0)
     {
       sfirsit = false;
       tasta = getch ();
       switch (tasta)
         {
           case 72: DSemn->Deplasare (0, -1);  // sus
                    break;
           case 80: DSemn->Deplasare (0, 1);  // jos
                    break;
           case 75: DSemn->Deplasare (-1, 0);// stinga
                    break;
           case 77: DSemn->Deplasare (1, 0);// dreapta
                    break;
           // nu e tasta cursor
           default: sfirsit = true;
         };
     };
```

```
301    } while (!sfirsit);
302    DSemn->Scriere ('=');
303    do
304    {
305     sfirsit = true;
306     tasta = getch ();
307     if (tasta == 0)
308       {
309        sfirsit = false;
310        tasta = getch ();
311        switch (tasta)
312          {
313          case 72: DSemn->Deformare (0, 1);
314                   break;
315          case 80: DSemn->Dilatare (0, 1);
316                   break;
317          case 75: DSemn->Deformare (1, 0);
318                   break;
319          case 77: DSemn->Dilatare (1, 0);
320                   break;
321          // nu e tasta cursor
322          default: sfirsit = true;
323        };
324     };
325    } while (!sfirsit);
326    DSemn->Stergere ();
327    delete DSemn;
328    getch ();
329    clrscr ();
330    return 0;
331  }
```

EX75.CPP

Constructori plus dinamică.

În listingul CODP2.CPP se lucrează cu streamurile versiunii 2.0. Acest lucru poate fi recunoscut după instrucţiunea de includere a FSTREAM.H în loc de STREAM.H. Deoarece compilatorul Borland C++ lucrează, în mod standard, cu biblioteca mai nouă a streamurilor, acest exemplu poate fi rulat direct din mediul de dezvoltare, fără un apel explicit al linkeditorului.

O altă particularitate a C++ este integrată în listingul CODP2.CPP: dacă e posibilă supraîncărcarea constructorilor, atunci de ce n -ar fi posibilă supraîncărcarea operatorilor? Acest lucru se întîmplă cu operatorul "put" ("<") al clasei "ostream". O astfel de supraîncărcare a operatorilor nu este altceva decît o redefinire a operatorului corespunzător. Pentru aceasta, în clasa corespunzătoare trebuie inserată o declaraţie "friend", care specifică că există o modificare în legătură cu operatorul. În continuare mai trebuie doar specificată noua funcţie a operatorului. Clasele "istream" şi "ostream" sînt clase derivate din clasa "ios". Prin expresia:

ios::app + ios::out

modul de deschidere al fişierului este fixat astfel încît fişierul este creat în cazul cînd nu există. Dacă există, se va scrie în el, în continuare, de la sfîrşit.

Cu aceasta, consideraţiile relative la constructori şi destructori se încheie.

Folosirea listelor în limbajul Prolog

(continuare din pag. 48)

Mecanismul de revenire este destul de contestat în lumea informaticienilor datorită consumului "inutil" de memorie şi timp, dar este mecanismul care ne garantează că, atunci cînd solicităm acest lucru, găsim TOATE soluţiile unei probleme, în mod automat. Acesta este aspectul care face din limbajul Prolog cel mai puternic instrument în demonstrarea automată a teoremelor şi în general a tuturor problemelor legate de inteligenţa artificială.

Mecanismul de revenire este puternic perturbat în activitatea sa de către predicatul "!" (cut). De aceea este bine să folosim cut-ul doar acolo unde sîntem absolut siguri că nu se pot genera soluţii utile. Un exemplu de folosire adecvată a acestui predicat poate fi regăsit în programul care măsoară viteza de sortare a unei liste, mai precis în cazul predicatului care generează lista de numere aleatoare. Dacă înlăturăm cut-ul dinaintea apelului recursiv"generez(C1,X)" singurul rezultat va fi apariţia mesajului de eroare referitor la umplerea stivei şi acesta doar dacă încercăm să generăm liste prea lungi (număr mare de iteraţii). Încercaţi să modifcaţi numărul de elemente care compun lista, înlocuind constanta "500" cu "50" şi îndepărtaţi cut-ul. Nu veţi sesiza nici o diferenţă în funcţionarea programului (cu excepţia lungimii listei şi a timpului). În schimb stiva consumată va fi mai mare.

În alte situaţii adăugarea unui "cut" poate determina programul să facă cu totul altceva decît ceea ce ne aşteptăm! Încercaţi să adăugaţi un "cut" înaintea lui "A<=X" din prima clauză a predicatului "sort1"din programul de sortare. Rezultatul va fi cu totul altul decît cel scontat în momentul conceperii predicatului! Pentru a avea date informative asupra stivei se poate folosi predicatul "*storage(Stack_size,Heap_size, Trail_size)*" care returnează dimensiunile corespunzătoare pentru cele trei zone de memorie menţionate. Dar aceasta este deja altă poveste.

Sper că din exemplele anterioare v-am convins de faptul că, deşi nu există nici un predicat care să prelucreze liste, putem face prelucrări complexe asupra acestora cu un efort destul de mic. Succes!

ing. Pantea Adrian

PC-uri extensibile

(continuare din pag. 32)

Suplimentar, trebuie luate în seamă şi elementele clasice de extensibilitate: capacitate maximă de memorie, număr maxim de drive-uri, etc.

Decizia de schimbare a microprocesorului trebuie luată după o analiză minuţioasă a necesităţilor aplicaţiilor pe care le rulaţi în mod curent. Practic trebuie să fiţi siguri că principala cauză a slabelor performanţe care v-au condus la necesitatea extinderii, se datoresc microprocesorului şi nu altor componente ale sistemului. De exemplu, dacă sînteţi nemulţumit de viteza Windows-ului, s-ar putea să obţineţi rezultate mai bune schimbînd adaptorul grafic decît microprocesorul.

În concluzie, se poate spune că PC-urile extensibile oferă, faţă de cele clasice, o şansă în plus în încercarea de a ţine pasul cu înnoirile tehnologice şi cu necesităţile crescînde ale aplicaţiilor. Această şansă este gratuită atîta timp cît preţul unui PC extensibil nu diferă de cel al unui PC clasic. Pentru cine se gîndeşte serios la posibilitatea extinderii microprocesorului in viitor, PC-urile extensibile oferă, în general, mai multe garanţii decît orice altă variantă, ele fiind proiectate cu posibilitatea extinderii procesorului în gînd.

Există totuşi şi riscul ca un utilizator care a cumpărat un PC extensibil să nu mai fie satisfăcut la un moment dat de variantele de extindere a procesorului oferite de fabricant, în măsura în care acestea nu ţin pasul cu avansurile tehnologice.

PC-urile extensibile introduc noi criterii în luarea deciziilor de achiziţionare a unui PC, fapt care poate fi considerat de unii ca o bătaie de cap suplimentară, iar de alţii ca o cale benefică de economisire a banilor de investiţii.

În partea a doua a articolului vom prezenta cîteva tipuri de PC-uri extensibile printre care şi cîteva care vor fi comercializate în ţară de firma DARIAN ROM SUISSE SRL.

Mitica Caraiani
DARIAN ROM SUISSE SRL,
tel. 95/123611, fax. 95/124567

Cînd cade reţeaua

Este ora 20^{15}, administratorul reţelei tocmai s-a aşezat pe canapea; se odihneşte ascultînd muzică clasică şi bînd un pahar de vin roşu. A trecut încă o zi încărcată pentru responsabilul principal cu sistemul de comunicaţii al întreprinderii.

Sună telefonul "Avem mari probleme", îi spune tehnicianul reţelei, "n-am vrut să vă deranjăm acasă, dar ştiţi şi dvs. cum este directorul şi dacă află, că datele i-au fost distruse, ... ştiţi şi dvs."

În toate întreprinderile care posedă un sistem informaţional şi de comunicaţii mare există lucrători care răspund de funcţionarea bună a acestora 24 ore din 24. Toţi pot relata despre evenimente care le-au răpit somnul. De cele mai multe ori acest fel de evenimente duc la modificări esenţiale în cadrul reţelei.

Cunoscînd experienţa altor administratori şi ştiind, cum anume au fost îndepărtate problemele apărute, se pot evita multe catastrofe.

Backup-ul perfect

Cu aproximativ un an în urmă s-a ajuns la un CRASH al server-ului principal dintr-o intreprindere internaţională de construcţii. Server-ul, conţinînd o bază de date de 2 GByte, plină cu informaţii importante, avea în componenţa sa un harddisk defect. Tocmai pentru un astfel de caz administratorul a fost pregătit şi cumpărase din timp un sistem de backup tip State-of-the-Art pe care a efectuat cu conştiinciozitate toate salvările de siguranţă. Iniţial această întîmplare nu a neliniştit pe nimeni, deoarece se ştia că un harddisk poate "să-şi dea sufletul" odată şi odată.

Deci a fost schimbat harddisk-ul şi testat. După ce server-ul a fost configurat din nou, s-a putut începe procedeul de restaurare. Dar spre surprinderea tuturor datelor salvate nu au putut fi restaurate. Administratorul tot încerca să copieze datele de pe sistemul de backup pe server. Dar degeaba. După 15 încercări succesive a telefonat producătorului acestui sistem. Abia atunci a ieşit la iveală faptul, că software-ul de backup pe care îl folosea era într-una din primele versiuni livrate, care conţinea o "scamă". Cu această versiune nu s-a putut face însă nici o restaurare.

Destul de puţini producători oferă hardware pentru Backup. Funcţionalitatea acestor aparate se aseamănă de obicei foarte mult. Cele mai mari diferenţe se pot observa la soft. Unele produse salvează doar datele, în timp ce altele salvează şi anumite tabele ale directoarelor sistemului de operare al reţelei cît şi funcţiile LAN. Dacă backup-ul este executat doar pentru date la o restaurare va trebui mai întîi configurat server-ul. Unele sisteme de backup permit utilizatorilor efectuarea de copii de siguranţă a datelor de pe PC-uri Desktop. În acest caz reţeaua este folosită ca şi mediu de transfer.

Sistemul de backup al întreprinderii de construcţii avea multe funcţii folositoare. El funcţiona perfect, cînd trebuia să salveze fişierele sistem şi de date. În schimb restaurarea n-a funcţionat, iar datele au fost pierdute pentru totdeauna. Nu este vorba aici doar de lansarea, de către producător, înainte de vreme a unui produs ci şi de faptul că administratorul s-a pripit. Este foarte bine că se introduc tehnologii noi, şi de regulă acest pas este răsplătit cu avantaje serioase în activitatea zilnică. Este însă foarte important să fie luate şi măsuri de precauţie atunci cînd se folosesc prima oară produse noi, care încă nu au fost testate în practică. În cazul de faţă ar fi trebuit să aibă loc o testare amănunţită a sistemului de backup înainte de a-l folosi efectiv. Ar fi fost de datoria administratorului să se convingă că sistemul nu doar salvează ci şi restaurează datele.

Tunete şi fulgere

Ce are în comun un jucător de golf, care se află singur în mijlocul terenului de golf, şi un server care se află lîngă geamul unui birou? Răspuns: Ambii pot fi loviţi de trăznet.

Acest lucru s-a întîmplat cu adevărat. Server-ul unui punct de lucru care se afla în exteriorul unei întreprinderi de comerţ se afla lîngă un geam deschis, şi într-o seară - a fost lovit de trăznet. În mod curios fluxul imens de energie a trecut prin server şi s-a descărcat în imprimanta laser cuplată la server. Aceasta s-a ars imediat.

A doua zi administratorul a fost aşteptat de utilizatorii miraţi, care îi raportau despre imprimanta carbonizată. Ca printr-o minune server-ul a trecut peste întîmplare doar cu un mic "şoc". Întreaga putere a trăznetului a afectat doar imprimanta, care nu a mai putut fi reparată.

De multe ori server-ele sînt tratate ca şi dulapurile din birouri. Ele se plasează într-un colţ, care întîmplător este liber, şi sînt folosite şi ca suport pentru flori. Nu trebuie uitat că aceste boxe sînt foarte importante pentru întreprindere şi mai ales că ele conţin foarte multe cunoştinţe.

Tot mai evidentă devine tendinţa de a centraliza reţeaua fizică. Aceasta se realizează prin îmbunătăţirea strategiei de cablare, a sistemelor de reţea şi a metodelor de management. Din ce în ce în mai multe locuri componentele hardware cele mai importante ale sistemului se păstrează în camere închise climatizate. Aici nu este vorba despre o revenire la "metodele mainframe", ci încet s-a ajuns la concluzia că aparatele LAN trebuie să fie protejate la fel ca şi bătrînul CALCULATOR. Este vorba doar de centralizarea reţelei fizice. Reţeaua logică, deci accesul la date, rămîne în continuare distribuită.

Nu doar datele valoroase, ci şi boxele, care le conţin, trebuie tratate cu grijă. În orice caz ar trebui folosită o sursă de tensiune neintreruptibilă. Dar şi aceasta trebuie testată mai întîi, scoţînd pur şi simplu stecker-ul din priză.

Server-ul dispărut

"În dimineaţa aceasta am intrat în reţea, şi unul dintre serverele noastre a dispărut. Este vorba despre serverul virtual de pe mainframe-ul DEC. Ştie cineva unde poate fi?"

Acest lucru s-a întîmplat într-adevăr într-o întreprindere în care un VAX

(DEC) era fost folosit în acelaşi timp atît ca HOST cît şi ca server virtual pentru LAN-ul cuplat.

Astfel inginerii şi administraţia puteau transfera date între aplicaţiile LAN şi HOST. Într-o dimineaţă utilizatorii au fost nevoiţi să observe că unităţile VAX-ului, care au fost repartizate LAN-ului, au dispărut. Nu s-au putut găsi nici datele nici sistemul de operare portabil LAN.

După ce nu s-a putut descoperi nimic stricat pe VAX, au fost verificate persoanele care s-au aflat în întreprindere la sfîrşitul săptămînii. S-a aflat astfel că în întreprindere a fost prezent şi un colaborator dintr-o altă întreprindere.

La acel sfîrşit de săptămînă sistemul de operare al VAX-ului a fost înlocuit cu o nouă versiune. Tehnicianul de service, care şi-a dat acordul , n-a ştiut că sistemul de operare LAN a fost instalat ca subsistem în cadrul sistemului de operare VAX. El a executat un simplu UPGRADE. În acest fel s-a scris peste toate sectoarele rezervate sistemului de operare LAN. Din fericire întreprinderea făcuse salvări regulate. Reinstalarea server-ului logic a durat însă mult timp şi l-au costat mulţi nervi pe administrator.

Sigur nu s-ar fi ajuns la scrierea peste LAN, dacă ar fi existat o documentaţie amănunţită. Chiar şi o atenţionare asupra caracteristicilor speciale ale acestui VAX ar fi fost de ajuns pentru a evita asemenea probleme. Acest lucru este valabil mai ales atunci cînd în sistem au fost implementate soluţii proprii.

LAN omoară pe Mainframe

"Ce a făcut LAN-ul dumneavoastră cu Mainfram-ul meu?" exclamă responsabilul minicalculatorului. "LAN-ul nu a stricat de fapt Mainframe-ul, doar controller-ul şi cca. 30 de utilizatori sînt afectaţi.."

O societate internaţională de comercializare a petrolului s-a hotărît să cumpere un Gateway 3270 la un preţ acceptabil. După ce Gateway-ul a fost instalat un întreg şir de utilizatori pasionaţi de LAN au încercat să facă transferuri de date către mainframe.

Controller-ul cluster-ului 3174, la care a fost legat Gateway-ul "şi-a dat duhul" în momentul în care nu a mai putut deservi toate cererile. Pe lîngă cererile din LAN s-au mai adăugat şi cele ale utilizatorilor de la terminale.

Urmarea: La scurt timp administratorul a primit un număr imens de telefoane de la utilizatorii mainframe supăraţi. Pentru eliminarea blocării a fost necesară iniţializarea, controller-ului. Ca urmare a acestei întîmplări poziţia ?????? de LAN din întreprindere a fost întărită şi administratorul de LAN şi-a pierdut credibilitatea.

La cumpărarea de LAN-uri sau de produse LAN-Host nu preţul trebuie să fie argumentul hotărîtor. Ca la multe alte produse, importantă este calitatea lor.

Cădere în ciuda toleranţei de eroare

S-a întîmplat într-o seară la ora 20^{45}. Telefonul a sunat: "Cred că ar fi mai bine să veniţi la birou", s-a comunicat administratorului. Ceea ce se întîmplase la întreprindere nu va putea fi uitat prea repede nici de conducere, nici de utilizatori şi nici de administrator.

În seara respectivă cîţiva colaboratori au lucrat pînă tîrziu cînd s-a constatat că reţeaua a "căzut". "Dar este posibil", a declarat un utilizator, care, a crezut pînă atunci că aplicaţia sa ar fi protejată prin toleranţa de eroare a LAN-ului. Un apel telefonic la secţia de suport al furnizorului sistemului confirmă: LAN avea toleranţă la eroare, şi totuşi a "căzut".

Ceea ce era mai rău în această situaţie nu era căderea sistemului, ci faptul, că această posibilitate nu a fost pusă în discuţie niciodată. Deci nimeni nu a fost pregătit pentru această eventualitate. În plus s-a constatat că competenţa furnizorului lasă de dorit. Au fost necesare mai mult de 3 zile pentru a găsi stricăciunea. Un controller de disc şi-a încetat activitatea. LAN-ul tolerant la erori nu dispunea pentru asigurarea datelor decît de o oglindire a unităţii şi nu şi de un duplicat al harddisk-ului. Pentru ambele unităţi nu exista decît un controller de disc şi o singură alimentare.

Deci controller-ul stricat a distrus datele de pe ambele unităţi.

Situaţia a devenit mai tensionată deoarece întreprinderea se afla într-o perioadă de economisiri: de aceea PC-uri ieftine avînd o capacitate de memorare mare au fost folosite ca servere. Astfel de aparate însă sînt fabricate doar de puţini producători. A durat 4 zile pînă cînd a putut fi găsit un nou controller.

Responsabilitatea reţelei nu poate fi dată niciodată producătorului produsului sau revînzătorului. Răspunderea pentru sistem este doar a întreprinderii şi deci a responsabilului de reţea. Trebuie create planuri pentru situaţii excepţionale, astfel încît oricînd să se poată rezolva problemele care apar.

Vise dulci

Cum se pot deci evita aceste coşmaruri?

O alternativă este să-ţi schimbi meseria. Dar atunci nu s-ar mai putea spune asemenea poveşti. Importantă este tratarea LAN-ului cît şi a server-ului cu aceeaşi grijă cu care s-a lucrat şi în lumea microcalculatoarelor. Multă lume încă mai consideră LAN-ul o jucărie, deoarece este relativ ieftin. Dar dacă LAN-ul a ajuns o componentă importantă a muncii zilnice, atunci trebuie luate aceleaşi măsuri riguroase, ca şi în cazul mainframe-ului.

Mai există şi o altă alternativă: dacă seara ne-am aşezat pentru a ne odihni pe canapea, atunci să decuplăm telefonul.

(M.I.)

*** Gratis * Gratis * Gratis ***

Pachet de rutine Pascal pentru derivare formală

Rutinele pot fi inserate în orice aplicaţie proprie!

Pentru aceasta nu trebuie să achitaţi decît contravaloarea dischetei şi a cheltuielilor de expediţie, sau să expediaţi o dischetă proprie, goală, pe adresa: Socaciu Ioan Tiberiu, str. Arieşului, nr. 102, bl. B2, sc II, ap. 55, 3400 Cluj-Napoca, tel. 95/140047.

*** Gratis * Gratis * Gratis ***

Şotii

Doriţi să faceţi o surpriză colegilor Dvs. îndrăgostiţi de Paradox? Introduceţi următorul program în Turbo Pascal şi îl instalaţi astfel (presupunînd că aveţi pe dischetă programul sub numele TEST.EXE):

```
C:\PDOX35> REN   PARADOX.EXE  PARADOX.OVR
C:\PDOX35> COPY  A:TEXT.EXE   PARADOX.EXE
```

Ce se va întîmpla la lansarea comenzii PARADOX ?

- se va încărca de fapt programul Dvs, care va pregăti mai întîi buffer-ul pentru zona tampon a tastaturii cu 2 caractere: "e" si "y" (codurile SCAN şi ASCII ale celor două caractere);
- se va executa o comandă REN PARADOX.OVR PARADOZ.EXE
- se va lansa în execuţie programul PARADOZ.EXE (care este de fapt adevăratul PARADOX);
- acesta nu va avea nici un parametru în linia de comanda (deoarece falsul PARADOX nu-i transmite nimic), astfel încît el va citi de la tastatura comenzile pe care urmează să le execute;
- deoarece în buffer-ul de tastatura sînt deja depozitate două caractere, programul se va conforma comenzilor primite ("Exit" şi "Yes") şi îşi va termina execuţia;
- se va executa o comandă REN PARADOZ.EXE PARADOX.OVR;
- în final va apare un mesaj care îl va pune pe gînduri pe operator!

```
{$M 1024,0,0} {$D-,L-,S-}
uses dos; var ss,zz:dirstr; fis:file;
begin memw[0:$41a]:=30; memw[0:$41c]:=34;
memw[0:$41e]:=4709; memw[0:$420]:=5497;
ss:=paramstr(0); zz:=ss; zz[0]:=chr(ord(zz[0])-3);
zz:=zz+'OVR'; ss[ord(ss[0])-4]:='Z';
assign(fis,zz); rename(fis,ss); exec(ss,'');
assign(fis,ss); rename(fis,zz);
writeln('Only God can help you !'); end.
```

mat. Nelu Cozac

Comenzi pentru grupuri de fişiere

Uneori am dori să executăm comenzi de tipul:

```
TYPE    *.PAS
```

O alternativă o oferă programul UFD. Astfel pentru a putea executa o "comandă" DOS

```
[d:][path]filename[.extension] afd
```

unde afd este un nume ambiguu (în care pot apărea joker-ii * şi ?), iar comanda poate fi executată numai cu nume neambigue, vom putea folosi cu succes comanda:

```
UFD[.BAT] [d:][path]filename[.extension] afd
```

"Secretul" constă în folosirea în fisierul de comenzi a unei linii de forma:

```
FOR %%f IN (%2) DO %1 %%f
```

Dar totul se mai poate complica: de ce să nu ştim pentru care fisier %%f se va executa comanda? Pentu aceasta vom folosi în fisierul de comenzi:

```
FOR %%f IN (%2) DO CALL UFFFFFFD %1 %%f
```

Sarcina fişierului UFFFFFFD.BAT este de a executa comanda DOS pentru un fişier specificat printr-un nume neambiguu şi de a afişa comanda. O soluţie ar putea fi:

```
rem ---UFFFFFFD.BAT---
echo %1 %2 > con
%1 %2
```

S-ar putea încerca şi afişarea prompt-ului:

```
rem ---UFFFFFFD.BAT---
echo on
%1 %2 > nul
@echo off
```

În acest caz în linia afişată apare şi "> nul". Oricum, afişarea corectă a prompt-ului este o temă interesantă. Şi acum conţinutul celor două fişiere de comenzi cu menţiunea că au fost testate sub MS-DOS 3.30 al firmei MicroSoft.

stud. Socaciu Ioan - Tiberiu, Cluj

```
------------ ufd.bat ----------------
echo .....................................
echo .        (C) SokySoft 1991          .
echo .   Writen by Socaciu Ioan - Tiberiu  .
echo .                                   .
echo .   UFD ver 1.1  ;  1991 - 08 - 14   .
echo . Adress: Ariesului 102/55, Cluj, Romania .
echo . Phone : 40 - 5 -140047 ( from abroad ) .
echo .       95 - 140047   ( from Romania ) .
echo .....................................
if /h==%1 goto help
if /H==%1 goto help
if ?==%1 goto help
for %%f in (%2) do call ufffffd %1 %%f
goto stop
:help
echo /-------------------------------------------------------------------------\
echo I Sintax: UFD [.BAT]  [ /h \ /H \ ? \ filename [ .EXE \ .COM \
.BAT ]  ufd ] I
echo \-------------------------------------------------------------------------/
:stop
echo on

rem ------------- ufffffd.bat -------------
echo %1 %2 >con
%1 %2
```

Deprotejarea fişierelor sub DR-DOS

Ca o replică la DOS-ul 3.30 al firmelor MicroSoft şi IBM a apărut DOS-ul firmei Digital Research (DR-DOS), o versiune elegantă de DOS, cu multe (chiar foarte multe) comenzi noi şi cele vechi cu mai multe opţiuni. O nouă comandă externă care apare este PASSWORD cu ajutorul căreia pot fi protejate prin parolă diverse fişiere. Parolele au o lungime de 8 caractere, astfel că utilizatorii pot folosi cu succes ca şi parolă seria de la buletinul de identitate.

Protecţiile pot fi la citire(r), scriere(w), ştergere(d), precum şi orice combinaţie a acestora. De exemplu protecţia completă este rwd. Pentru alte detalii, precum şi pentru sintaxă nu aveţi decît să tastaţi PASSWORD /H.

S-ar putea să fi uitat parola, sau (de ca să nu o spunem pe şleau: tentaţia copierii fişierului colegului este mare - de genul celei de a mînca mere interzise în grădina biblică ...). Pentru aceasta trebuie să ne amintim de structura intrărilor într-un director: 32 de octeţi cu următoarea semnificaţie:

```
00h-07h numele fişierului
08h-0ah extensia fişierului
0bh     atributele fişierului: (pe biţi: normal, protejat,
        scriere, ascuns, sistem, etichetă volum, subdirector,
        arhivat)
0ch-15h rezervaţi
16h-17h ora ultimei modificări
18h-19h data ultimei modificări
1ah-1bh prima unitate de alocare
1ch-20h dimensiunea fişierului
```

Ei bine, PASSWORD marchează diversele protecţii şi informaţiile despre parolă pe octeţii rezervaţi. Este de ajuns să puneţi 00h pe octeţii 14h si 15h intrînd cu PCTOOLS, NU sau orice alt utilitar de lucru la nivelul discului. De asemenea vă invităm să scrieţi un program de deprotejare ţinînd cont că există deja scrise rutine de citire şi scriere sector (vezi "if 6/91, pp52-53) sau eventual dacă doriţi să vă scrieţi propriile dumneavoastră rutine folosiţi întreruperile 25h şi 26h. Succes!

stud. Socaciu Ioan - Tiberiu, Cluj

Paza bună ...

Se întîmplă uneori ca, după o funcţionare normală a calculatorului, să apară la consolă un mesaj de tipul:

"Invalid drive specification",

atunci cînd încercăm să accesăm una din partiţiile discului hard. Este un semn că tabela de partiţii (Master Boot Record) a fost alterată, şi pentru marea majoritate a utilizatorilor singura ieşire o reprezintă reformatarea discului şi inevitabila pierdere a tuturor informaţiilor conţinute de acesta.

De asemeni, în urma modificării accidentale a parametrilor din SETUP (tip disc, configuraţia perifericelor), este posibil să apară mesaje de genul:

"Non system-disk or disk error", sau

"CRC mismatch in CMOS", etc.

Programul pe care vi-l propun permite salvarea tabelei de partiţii şi a informaţiilor din CMOS, şi restaurarea în caz de necesitate a acestor informaţii. Prin refacerea tabelei de partiţii, el va distruge automat şi orice virus de boot apărut între timp.

Pentru a obţine un program executabil se poate folosi TurboAsm, după cum urmează:

C:\MYDIR>TASM /Z/T XXX

C:\MYDIR>TLINK /T/X XXX

Va fi creat executabilul XXX.COM. În lipsa programelor Borland putem folosi LINK-ul de la Microsoft, apoi programul EXE2BIN:

C:\MYDIR>MASM XXX

C:\MYDIR>LINK XXX

C:\MYDIR>EXE2BIN XXX.EXE XXX.COM

În continuare, vom formata o dischetă sistem, apoi vom copia programul XXX.COM pe dischetă. Vom lansa programul cu opţiunea pentru salvarea informaţiilor (vom tasta '1'). Pe dischetă vor fi create două fişiere, CMOS.SAV şi MBR.SAV. Vom proteja discheta la scriere, şi în caz de necesitate vom lansa sistemul de pe această dischetă şi vom executa XXX, cu opţiunea de restaurare (vom tasta '2').

Cristian Paris

```
;
; Salveaza/reface informatiile din CMOS si MBR
;
;---------- CONST DEF ----
CR      equ     0dh
LF      equ     0ah
WR      equ     3
RD      equ     2
WRITE   equ     40h
READ    equ     3fh
;---------- MACRO DEF ----
print   macro   string
        lea     dx,string
        mov     ah,9
        int     21h
        endm
input   macro
        mov     ah,8
        int     21h
        endm
create  macro   name
        lea     dx,name
        xor     cx,cx
        mov     ah,3ch
        int     21h
        endm
open    macro   name
        lea     dx,name
        mov     ax,3d02h
        int     21h
        endm
file    macro   transfer,n_bytes,buffer
        lea     dx,buffer
        mov     cx,n_bytes
        mov     ah,transfer
        int     21h
        endm
close   macro
        mov     ah,3eh
        int     21h
        endm
sector  macro   transfer,buffer
        mov     ah,transfer
        mov     al,1
        mov     dx,80h
        mov     cx,1
        lea     bx,buffer
        int     13h
        endm
cseg    segment byte 'code'
        assume  cs:cseg,ds:cseg,es:cseg,ss:nothing
        org     100h
start:  jmp     entry
;---------- DATA ---------
mbr     db      "a:\mbr.sav",0
cmos    db      "a:\cmos.sav",0
buff    db      200h dup( ? )
mess1   db      CR,LF,"Tastati:",CR,LF," 1 - pentru salvare",CR,LF
        db      " 2 - pentru restaurare",CR,LF,"$"
mess2   db      "Scoateti discheta si apasati orice tasta "
        db      "pentru resetare...",7,"$"
reset   label   dword
        dw      0
        dw      0ffffh
;---------- CODE ---------
entry:  print   mess1
        input
        cmp     al,'1'
        je      short save
        cmp     al,'2'
        jne     entry
        jmp     restore
;
; Cod pentru salvare
;
save:   lea     di,buff     ; salveaza configuratie CMOS
        mov     ax,10h
        cld
again:  out     70h,al
        inc     ax
        push    ax
        in      al,71h
        stosb
        pop     ax
        cmp     ax,34h
        jne     again
        create  cmos        ; scrie in fisierul CMOS.SAV
        mov     bx,ax       ; handle
        file    WRITE,23h,buff
        close
        sector  RD,buff     ; citeste Master Boot Record
        create  mbr         ; scrie in fisierul MBR.SAV
        mov     bx,ax
        file    WRITE,200h,buff
        close
done:   ret                 ; gata
;
; Cod pentru restaurare
;
restore:
        open    cmos        ; citeste fisierul CMOS.SAV
        jc      done        ; poate fis. nu exista
        mov     bx,ax
        file    READ,23h,buff
        close
        lea     si,buff     ; reface informatiile din CMOS
        mov     ax,10h
        cld
again1: out     70h,al
        inc     ax
        push    ax
        lodsb
        out     71h,al
        pop     ax
        cmp     ax,34h
        jne     again1
        open    mbr         ; citeste fisierul MBR.SAV
        jc      done
        mov     bx,ax
        file    READ,200h,buff
        close
        sector  WR,buff     ; scrie tabela de partitii
        print   mess2
        input
        jmp     reset
cseg    ends
```

XXX.ASM

Uite mouse-ul şi cînd nu e ...

Programul pe care vi-l prezentăm vine din concursul de '1024' al revistei DOS International.

Programul rămine rezident în memorie la primul apel. Orice apel ulterior mai consumă încă 627 octeţi. Mousetsr face ca aplicaţiile care în mod normal nu lucrează cu mouse-ul să devină dintr-o dată operabile cu ajutorul acestuia. Principiul de funcţionare este pe cît de simplu pe atît de genial: mişcările mouse-ului şi apăsarea tastelor acestuia sînt "traduse" în apăsări de taste şi înscrise în bufferul de tastatură. Mousetsr transformă mişcările mouse-ului în taste cursor; în plus, tasta stîngă a mouse-ului simulează tasta ENTER, tasta dreaptă a mouse-ului simulează tasta ESC. În felul acesta, aproape toate programele operabile via meniuri pot fi deservite via mouse, chiar dacă programele respective nu sprijină lucrul cu mouse-ul. Asamblaţi "Mousetsr.asm" pur şi simplu cu TASM, linkeditaţi apoi cu TLINK cu opţiunea /t. Obţineţi astfel un fişier .COM bun de lucru (lung de 227 octeţi !!).

Gerd Knorr

```
1       ; program: mousetsr.asm
2       ; functie: emulare mouse
3       ; limbaje: TASM, TLINK /t
4       ; autor:  Gerd Knorr
5       ; (c) 1991 DMV Widuch GmbH & Co.Kg
6       title Mouse-Emulator
7       model     tiny
8       code      segment
9                 assume cs:code
10                org 100h
11      start:
12                jmp init
13      activ     db 0
14      event_handle:
15                push es; salvare registrii
16                push ax
17                push bx
18                push cx
19                push dx
20                pushf
21                cmp cs:activ, 1          ; deja activ? afara!
22                jz sf2
23                mov cs:activ, 1 ; seteaza activ
24                mov bx,ax
25                and ax,8h  ; tasta mouse dreapta?
26                jz next1   ; nu, continua
27                mov ax, 11bh             ; da, ESC
28                call write_char
29                jmp sf1
30      next1:    mov ax,bx
31                and ax,2h  ; tasta mouse stinga?
32                jz next2   ; nu, continua
33                mov ax, 1c0dh            ; da, ENTER
34                call write_char
35                jmp sf1
36      next2:    cmp cx, 320-8            ; spre stinga?
37                jnc next3
38                mov ax,4b00h
39                call write_char
40                jmp reset
41      next3:    cmp cx, 320+8            ; spre dreapta?
42                jbe next4
43                mov ax,4d00h
44                call write_char
45                jmp reset
46      next4:    cmp dx, 96-8             ; sus?
47                jnc next5
48                mov ax,4800h
49                call write_char
50                jmp reset
51      next5:    cmp dx, 96+8             ; in jos?
52                jbe sf1
53                mov ax,5000h
54                call write_char
55      reset:    mov ax,04h               ; cursor mouse la mijloc
56                mov cx,320
57                mov dx,96
58                int 33h
59      sf1:      mov cs:activ,0           ; reseteaza activ
60      sf2:      popf                     ; restaurare registrii
61                pop dx
62                pop cx
63                pop bx
64                pop ax
65                pop es
66                retf                     ; gata, iesire
67      write_char proc near
68                mov bx,0040h
69                mov es,bx
70                mov cx,es:[1ah]          ; cx = next
71                mov dx,es:[1ch]          ; dx = last
72                add dx,2h
73                cmp dx,3eh               ; sfirsit?
74                jnz wr1                  ; nu, continua
75                mov dx,1eh               ; da, revine la inceput
76      wr1:      cmp cx,dx  ; plin?
77                jnz wr2                  ; nu, continua
78                ret                      ; da, return
79      wr2:      mov bx,es:[1ch]
80                mov es:[bx],ax           ; pune cod tasta
81                mov es:[1ch],dx          ; actualizare last
82                ret
83      write_char endp
84      init:
85                mov ax,0h  ; initializare driver mouse
86                int 33h
87                cmp ax,0h  ; driver de mouse instalat?
88                jz norm                  ; nu, sfirsit normal
89                mov ax,0fh ; modifica viteza
90                mov cx, 50 ; orizontal
91                mov dx, 50 ; vertical
92                int 33h
93                mov ax,0ch; instalare event_handler
94                mov cx,0bh               ; apel miscare, dreapta, stinga
95                mov dx,ds  ; adresa segment
96                mov es,dx
97                mov dx,offset event_handle
98                int 33h
99                mov ax,3100h
100               mov dx,001ch
101               int 21h
102     norm:     mov ax,4c00h
103               int 21h
104     code      ends
105     end       start
106
```

MOUSETSR.ASM

Ştiaţi că...

- pînă şi cele mai ieftine dischete de 3,5" DD se pot formata şi folosi la 1,44 Mo, practicînd o gaură de diametru cel puţin 4,5mm, simetrică cu cea existentă din fabricaţie?
- dischete îndoite pot fi eventual totuşi citite dacă le puneţi în prealabil într-un plic nou, pe care îl tăiaţi cu grijă de pe o dischetă de sacrificiu?
- dischete pe care s-a vărsat cafea mai pot fi eventual recuperate, dacă le introduceţi într-un plic uscat, ca mai sus?
- la copierea unor pachete soft de pe un harddisk este indicat să folosiţi comanda "xcopy /e/s", pentru a prelua şi directoare vide, în absenţa cărora anumite programe nu lucrează?
- puteţi "bloca" dischete cu o bucăţică micuţă de bandă adezivă şi o pensetă? Cine ar vrea să "studieze" discheta care vă aparţine nu obţine decît "Read Error"...

Cod	Nume pachet	Nr. dischete	Format	Capacitate	Limba
Sisteme de operare					
29	DR - DOS 6.0	1	3,5"	1,4M	E
Cataloage					
1	Netware Buyer's Guide	2	5,25"	1,2M	E
20	PD Servicelage Katalog	2	5,25"	360K	D
21	DS - Katalog	2	5,25"	360K	D
22	Camuflage-Soft Katalog	1	5,25"	1,2M	D
23	Freeware Service Katalog	2	5,25"	360K	D
Aplicaţii					
12	CAPP	1	5,25"	1,2M	D
2	Buch,Trio	2	5,25"	360K	D
39	BALPRO	1	5,25"	360K	R
Utilitare, biblioteci					
14	Borland Windows Tools	1	5,25"	360K	D
17	KAO BTX-Dekoder + Demo	1	5,25"	360K	D
36	QuickStep Tools	2	5,25"	1,2M	D
6	Spindrift Libraries Demo	1	5,25"	360K	E
7	BKS Demo	1	5,25"	360K	D
38	List v 3.0	1	5,25"	360K	R
Baze de date					
10	Paradox 3.0	1	5,25"	360K	D
13	Foxpro 1.02	2	5,25"	360K	D
31	Foxpro 2.0	2	3,5"	720K	D
18	SOS - Software Tools Datenbank	1	5,25"	360K	D
24	Advanced Revelation	1	3,5"	720K	E
25	Rbase 3.1	1	3,5"	720K	D
32	WindowBase	1	5,25"	1,2M	D
33	Address One	1	5,25"	360K	D
37	Superbase 4	2	3,5"	720K	D
Grafică, DTP, CAD					
11	Time Works Publisher	2	5,25"	360K	D
15	MEGA CAD	1	5,25"	360K	D
27	Aldus Page Maker 4.0	1	3,5"	720K	E
30	Draw Perfect 1.1 si Office 3.0	1	3,5"	1,4M	D
9	Micrografx	1	5,25"	360K	D
Cursuri					
5	DOS - Curs IBM	1	5,25"	360K	D
Procesoare de texte, OCR					
28	Word Perfect for Windows	1	3,5"	1,4M	D
35	Scout OCR	1	5,25"	360K	D
40	Ekta Editor V1.5	1	5,25"	360K	E
Măsură şi control					
3	Labtech Notebook Demo	1	5,25"	1,2M	E
4	Labtech Control	1	5,25"	1,2M	E
34	Scope	1	5,25"	360K	D
Jocuri, diverse					
8	Pearl Agency	1	5,25"	360K	D

Un nou serviciu pentru cititorii noştri:

SHARE - if

Pentru o mai bună circulaţie a informaţiei între ofertanţi şi solicitanţi de programe (şi nu numai ...), vă oferim posibilitatea să:

- economisiţi timpul necesar introducerii programelor mai lungi, al căror listing a apărut în **"if"**.
- preluaţi pe dischetă, pentru eventuale multiplicări, textul articolele din **"if"** pe care aţi vrea să le aveţi disponibile şi în această formă
- preluaţi acele programe şi texte de articole care ne-au fost trimise spre publicare, dar pe care, din variate motive (cel mai adesea, lipsa de spaţiu) - nu ajungem să le publicăm. (Desigur, aceste materiale le difuzăm numai atunci cînd există acordul autorului - încă nu am găsit o variantă de recompensare a autorului pentru această variantă de difuzare) O listă incompletă: criptare fişiere, completare formulare, o parolă solidă, nucleu de multitasking, "catalog" de dischete, etc.
- faceţi cunoştinţă cu acele programe din care ni s-au trimis demo-uri, respectiv **să difuzaţi programele dumneavoastră demonstrative** prin intermedierea noastră. Acest din urmă punct sperăm să intereseze în mod deosebit pe toţi autorii noştri de programe, cărora le este destul de greu să-şi facă publicitatea de care au, evident, nevoie.

Cei dornici să includem demo-urile lor în oferta noastră sînt rugaţi să achite, odată cu expedierea dischetei sursă, suma de 500 lei în contul nostru: Cont nr. 40729-96010402, Banca Română pt. Dezvoltare Mureş.

(Cei care trimit materiale informativ/didactice, şi nu publicitare, nu au de achitat nimic.) Preţul pe care îl practicăm pentru o dischetă este, din păcate, greu de definit în avans, din motive cunoscute de toată lumea. De aceea vă rugăm să vă interesaţi - telefonic sau în scris - în momentul lansării comenzii. În principiu, preţurile sînt identice cu preţurile cu care, la rîndul nostru, am achiziţionat dischetele, la care se adaugă 50 lei pentru munca noastră şi, suplimentar, taxa de expediere. Desigur, cei care ne trimit dischetele lor, au de achitat doar taxa de copiere şi expediere, iar cei care trec sau trimit pe cineva pe la redacţie, vor avea de achitat doar cei 50 lei, taxa de copiere.

Vă rugăm ca pe comanda pe care ne-o trimiteţi să specificaţi pe lîngă numele pachetului şi capacitatea şi formatul dischetelor pe care doriţi să vă furnizăm produsul.

Revista presei

Cu toate dificultăţile pe care le implică tipărirea unei reviste, sau cărţi de specialitate, greutăţi pe care le ştim, vai, atît de bine, ne bucură faptul că (şi) iformaticienii scriu şi că numărul publicaţiilor de informatică este în creştere. Ne face plăcere să vă prezentăm cîteva din ultimele apariţii sosite la redacţie. Pentru a afla dacă între timp n-au mai apărut şi alte numere, pentru informaţii suplimentare şi comenzi vă rugăm să vă adresaţi direct acestor redacţii.

CHIP - este o revistă de calculatoare personale, editată la Braşov sub licenţa editurii Vogel Verlag din Germania. Tipărit în condiţii grafice excelente, numărul 2 al revistei (decembrie 1991), are 108 pagini şi costă 198 lei. Din cuprinsul acestui număr ne-au reţinut în mod deosebit atenţia testele hard (calculatoare 486SX, monitoare, AT-uri 286 şi imprimante laser), prezentarea sistemului de operare Unix şi Povestea primului PC.

CHIP, D. Gherea 19, 2200 Braşov, tel. 921/18643, fax 921/50012.

Computerland Bucureşti Magazin este o revistă de firmă editată de Computerland Bucureşti. Numărul 3 al revistei (februarie-martie 1992) are 16 pagini A3. Revista este distribuită prin intermediul reţelei de magazine proprii. Pe lîngă rubricile Buyer's guide, în cadrul cărora sînt prezentate produsele comercializate în reţeaua de magazine Computerland, vă recomandăm să citiţi articolele: Reţele locale de tip Novell realizate cu calculatoare MBL, Dialog AutoCAD, Baze de cunoştinţe, Interfaţa grafică Apple.

ComputerLand Bucureşti, Calea Dorobanţilor, nr. 152, Bucureşti, tel. 333227, fax 127568.

Infoclub - revistă de informatică şi calculatoare editată de Societatea Ştiinţă şi Tehnică S.A. în colaborare cu International Data Group. Selectăm cîteva articole din numărul 4/1991 (48 pagini, 49 lei): Cucerind lumea cu fax-ul şi telefonul, Ziua în care vin... PC-urile, Antivirusurile şi securitatea informaţiilor, SQL Server şi Lan Manager, Curs MS-DOS.

Infoclub, Piaţa Presei Libere nr. 1, Bucureşti, cod 79781, tel. 176010/1151.

Hello CAD Fans - Revistă de proiectare asistată de calculator editată de Fast Impex Ltd. Bucureşti. În numărul 7, ianuarie 1992 (32 pagini, 50 lei) puteţi citi articolele: Autodesk intră decisiv în arenă, Haşurare şi cotare, AutoCAD Release 11 - un binevenit pas înainte, AutoLISP? Take it easy..., Formate, scări şi biblioteci parametrizate.

Hello CAD Fans, CP 39-119, Bucureşti, tel. 90/534398

Pro Informatica - revistă editată la Cluj de Microinformatica SRL. Pe lîngă rubrici interesante pentru utilizatorii de PC-uri: Tehnica imprimării color, Unix la locul şi timpul potrivit, DOS fără documentaţie, numărul 1 al revistei (52 pagini, 80 lei) mai conţine o secţiune întitulată Informagica (Fractali, Black holes-uri în informatică, Biliard grafic) şi articole destinate cititorilor pasionaţi de calculatoarele compatibile ZX Spectrum.

Microinformatica S.R.L., str. Observatorului 1, 3400 Cluj, tel./fax 95/118263.

hobBIT - revista este editată de hobBIT S.N.C. Bucureşti şi este destinată amatorilor de calculatoare Spectrum, HC, Cobra, Jet, Commodore 64/128, IBM-PC. Numărul 2/1992 (32 pagini, 70 lei) conţine articolele: Spectrum Forth, Commodore 6502/6510 - Limbaj maşină, Wing Commander, Dizzy, Tips & Tricks. Tot firma hobBIT a editat recent un interesant Almanah, destinat posesorilor de calculatoare personale tip Spectrum, Commodore şi IBM (200 lei).

hobBIT S.N.C., C.P. 37-131, Bucureşti.

Talon de comandă-abonament

Subsemnat(a/ul).....
doresc să-mi trimiteţi |_____| exemplare din fiecare număr al revistei **if**, începînd cu numărul |_____| , pe adresa:
.....
.....
.....
Telefon: acasă..... la servici.....
Pentru aceasta am expediat prin mandat poştal/virament/cec.....lei în contul nr. 40729-96010402 deschis la Banca Română pt. Dezvoltare Mureş. Abonamentul funcţionează pînă la epuizarea sumei; în cazul că banii pe care i-am trimis nu mai ajung pentru un alt număr, voi fi anunţat de acest lucru în timp util, astfel încît nu risc să pierd nici un număr. Dacă mă răzgîndesc şi renunţ la abonament, mi se returnează toată suma rămasă.

if 1 / 92 **Data:** ..... **Semnătura** ..... _

Important !

În curînd va apare cartea

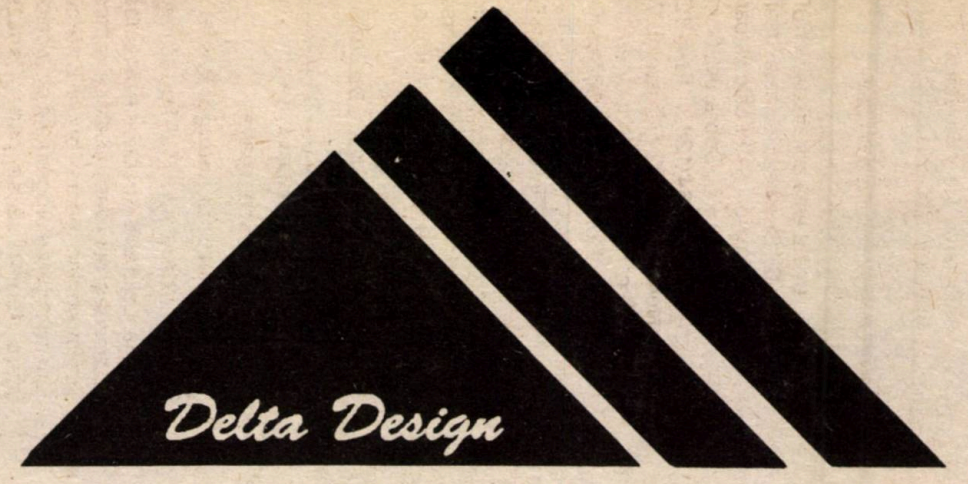
Programarea în limbajul Pascal

editată de Micro ATCI Tîrgu-Mureş.

Manualul este o traducere din limba germană şi este destinat atît celor care vin în contact pentru prima dată cu acest limbaj de programare cît şi celor care doresc să se perfecţioneze învăţînd noi metode şi tehnici de programare. Volumul este bogat în exemple, toate programele prezentate fiind încercate şi testate cu ajutorul compilatorului Turbo Pascal 6.0 al firmei Borland.

Tirajul fiind limitat, asiguraţi-vă că nu veţi pierde lucrarea expediind o comandă fermă pe adresa:

Micro ATCI, C.P. 64, 4300 Tîrgu-Mureş



Delta Design S.A., Hotel Bucureşti, 63-81 Victoria Street,
Wing D - Agencies, Stair E 2, 4th Floor, Sect. 1, Bucharest,
Phones: 15.42.26 / 40 015.45.80 Ext.2404 / Fax: 40 13.60.40.

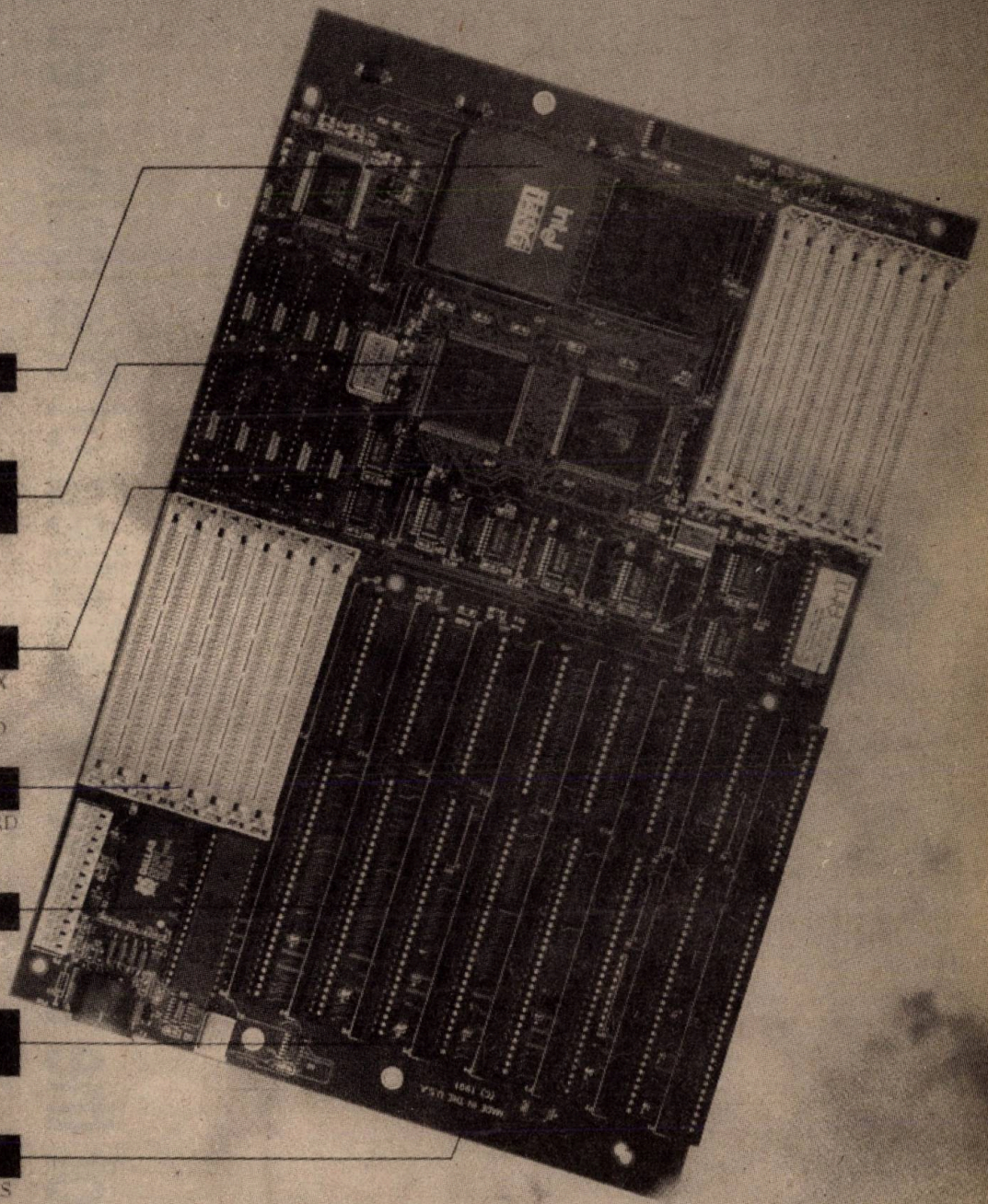
99.94% of the users fully satisfied of the delivered products

Authorised distributors for the following companies:

Tulip Computers, Apple Computer Inc., Microsoft, Autodesk,
Ashton Tate, Claris, Corel, Delta Design International etc.

ABTECH
486-33s CACHE

STEP INTO THE FUTURE



PROCESSOR
80486-33MHz

CACHE CONTROLLER
SYMPHONY 82C 465
UP TO 265 KB CACHE

CHIP SET
SYMPHONY 82C 460DX
HIGHLY RELIABLE &
HIGHLY INTEGRATED

MEMORY
UP TO 64MB ON BOARD

BIOS
AMI / PHOENIX

EXPANSION SLOT
EIGHT 16-BIT SLOTS

BABY BOARD
FITS DIFFERENT CASES

ABTECH, str. Pitar Moş 2-4, sector 1, Bucureşti, tel. 122784, 148170

CALCULATOARE
PERSONALE

Revistă editată de Micro ATCI S.R.L. Tîrgu Mureş

3 / 92
(14)
150 Lei
ISSN 1220-1529

Baze de date:

- Baze de date orientate obiect
- BLOB - Binary Large Objects
- Baze de date Client-Server

» Ce calculatoare putem cumpăra, de unde şi la ce preţuri?

» PC-uri extensibile

» Instalarea reţelelor Novell

Cursuri:

- C++ (IV)
- Programarea Windows (I)

Ultima oră!

S-a înfiinţat POLICOM S.A., societate comercială multinaţională cu capital privat, avînd principal acţionar Ab Mod S.N.C.

Noua firmă vă oferă un sortiment larg de produse:

- Calculatoare personale compatibile IBM
- Imprimante matriciale
- Dispozitive grafice
- Floppy - dischete
- Produse software originale
- Copiatoare Toshiba
- Truse şi accesorii pentru întreţinere

Nu uitaţi!

POLICOM S.A. un partener potenţial pentru afacerile dumneavoastră.

CLUJ: B-dul 22 Decembrie nr. 135 tel. 95-156350

Cuprins

Hotline

- Apple în România . . . . . pag. 5
- IBM îşi extinde familiile de calculatoare PS/2 şi PS/1 . . . . . pag. 6
- ROMORGADATA '92 . . . . . pag. 6
- CERF '92 . . . . . pag. 6
- Puterea unei workstations la nivel DOS . . pag. 8

News

A picta ca van Gogh . . . . . pag. 9

Alpha - o nouă tehnologie a procesoarelor . . pag. 10

Tehnologii

Head Up Display . . . . . pag. 12

Fundamente

Comunicarea om-maşină . . . . . pag. 14

Sisteme multimedia. Metode de acces la text. Metoda semnăturii. . . . . pag. 38

Vă prezentăm

Construirea de sisteme multimedia . . . . . pag. 20

- Noutăţi la Guru 3.0 . . . . . pag. 21
- Sisteme expert create cu Guru . . . . . pag. 21

Baze de date

Blob, blob... . . . . pag. 22

Baze de date orientate obiect . . . . . pag. 23

Baze de date Client-Server . . . . . pag. 25

- Gupta SQLBase Server . . . . . pag. 27
- Informix SE şi OnLine . . . . . pag. 28
- Ingres SQL-Server . . . . . pag. 28
- Oracle SQL-Server . . . . . pag. 28
- Sybase SQL-Server . . . . . pag. 29
- XDB-SQL-Server . . . . . pag. 29
- Novell NetWare SQL . . . . . pag. 29

Anchetă

Piaţa românească de calculatoare . . . . . pag. 34

Hardware

PC-uri extensibile (2) . . . . . pag. 36

Cursuri

Stăpînind ierarhii, Curs C++ (IV) . . . . . pag. 44

Programarea Windows (I) . . . . . pag. 49

Reţele

Instalarea sistemului de operare Netware 2.2 (1) . . . . . pag. 54

Laborator

Cîte ceva despre VGA . . . . . pag. 58

- Organizarea memoriei grafice la VGA. . pag. 58
- Programarea paletei de culori pentru modurile cu 256 culori. . . . . pag. 58
- Modul 320x240x256 . . . . . pag. 58
- De ce am ales modul 320x240x256 ? . . pag. 58
- Modul 360x480x256 . . . . . pag. 58

Colţul programatorului

Cine indexează mai bine ? . . . . . pag. 60

- Clipper 5.01 . . . . . pag. 60

Practica

- Trucuri Quattro Pro 3.0 . . . . . pag. 61
- Corecţii Borland în unit-ul »editors« (Turbo Vision) . . . . . pag. 62
- Cod autoreproducător . . . . . pag. 62
- Here comes the music ... . pag. 63
- Reversi - 64:0 . . . . . pag. 63
- În contra viruşilor . . . . . pag. 64
- Un doctor pentru dischete . . . . . pag. 64
- Print - o comandă ignorată? . . . . . pag. 64

Rubrici

- Caseta redacţiei . . . . . pag. 4
- Editorial . . . . . pag. 4
- SHARE - if . . . . . pag. 65

Aşteptîndu-l pe Godot

Luna mai a fost o lună bogată în expoziţii de tehnică de calcul: ROMOR-GADATA, la Pavilionul Expoziţional, CERF '92, la Polivalentă, Expoziţia Apple, la Sala AGIR. Expoziţiile s-au constituit, toate în felul lor, în adevărate evenimente pentru împătimiţii într-ale calculatoarelor, care-au avut ocazia să-şi scalde cel puţin privirea, dacă punga nu le-a permis mai mult. Din păcate chiar şi cei cu punga plină au rămas, de data aceasta, numai cu privitul. Ceea ce a caracterizat, mai mult sau mai puţin, toate aceste expoziţii a fost aşteptarea. Numai că cel aşteptat nu era Godot ci Fixingul. Şi fixingul nu venea... Nici din vina expozanţilor şi nici din cea a organizatorilor expoziţiilor. Vina era în altă parte şi nu era o vină colectivă. Aşa că a trebuit să ne mulţumim să privim, să mirosim şi eventual să pipăim. Ne bucură în schimb faptul că multe produse interesante, lansate nu de mult pe alte pieţe, au apărut şi în ţară. Aş enumera doar cîteva: Windows 3.1, C++ 3.0, Object Vision 2.0, Ventura Publisher 4.0, 486/50, Screen Machine.

Expoziţia de la Polivalentă a fost interesantă şi prin încercarea de specializare a profilului, calculatoarele ne-mai-fiind expuse printre articole de lenjerie şi sticle cu sucuri. Reuşita a fost doar parţială nu atît pentru faptul că au fost expuse şi produse electronice, fapt inclus de altfel în program, cît datorită faptului că multe firme importante au lipsit de la manifestare. Sperînd că se va reuşi trecerea peste relaţiile de "familie", noi aşteptăm în continuare primul Comdex sau CeBIT românesc. Oricum ne bucură faptul că, de această dată, sub cupola sălii, în loc să sclipească lămpaşele garbage-collector-ilor Pieţei Universităţii au sclipit monitoarele. Mulţumim!

Ne cerem scuze pentru ţinuta numărului anterior. Purtînd numărul 13, a avut ghinion. Încercăm să revenim cu o ţinută mai decentă, chiar dacă preţul este mai ridicat. Este important conţinutul, dar se mai întîmplă şi ca haina să-l facă pe om.

Pentru cititorii care nu au reuşit să facă cunoştinţă cu revista de la început, anunţăm că revista a intrat într-al treilea an, că au apărut 3 numere în 1990 (1-3/90), 8 numere în 1991 (1-8/91) şi, cu acesta, 3 numere în 1992 (1-3/92). În stoc mai sînt disponibile reviste din toate numerele apărute în '91 şi '92, mai puţin numărul 3/91. Din '90 nu mai avem în stoc nici o revistă.

În acest număr intenţionam să vă prezentăm un amplu material de sinteză despre piaţa românească de calculatoare personale şi de dischete. Întrucît am primit foarte puţine răspunsuri pînă la închiderea ediţiei, sinteza obţinută este mai subţire. Deocamdată nu vă putem spune nimic despre dischete (la CERF o dischetă bună costa 500-600 lei). Pe măsură ce vom intra în posesia unor noi informaţii vom face şi noi un "refresh". De asemenea îi rugăm pe toţi cei care doresc să participe la anchetele pe care le vom iniţia în viitor să ne scrie, specificînd domeniile de interes.

Anunţăm cititorii că nu am renunţat la nici unul din concursurile lansate. Aşteptăm răspunsurile Dvs. Promitem ca de îndată ce vom intra în posesia cel puţin a unei bancnote de 5.000 lei, să mărim cel puţin premiul concursului "1024 de lei pentru cel mai bun program de cel mult 1024 de octeţi".

În acest număr începem un nou curs: "Programarea Windows" şi o serie de articole despre instalarea reţelelor Novell. De asemenea, am acordat un spaţiu mai larg problematicii bazelor de date, prezentînd principalele S.G.B.D.-uri client-server. Chiar dacă aceastea au pătruns încă prea puţin în ţară, subiectul este interesant, şi întreprinderile medii şi mari nu se vor putea descurca în viitor fără un astfel de S.G.B.D. Aşa că este bine să ştim ce ne aşteaptă, măcar din acest punct de vedere...

Romulus Maier

if

revistă de informatică
editată de firma **Micro ATCI**

Director: ing. Dumitru Dunca

Redacţia:
ing. Iosif Fettich,
ing. Ingrid Maier,
ing. Romulus Maier,

Colaboratori externi:
ing. Caraiani Mitică
mat. Cozac Nelu
ing. Guţică Mirela
Hamor Soft SRL
ing. Malide Cristian
şef lucr. ing. Marinceu Dimitrie
corectura art. "Comunicarea om-maşină" a fost făcută de dna. prof. Sârbu Nastasia
cursul "Programarea Windows" a fost revizuit de:
ing. Pop Adrian şi
mat. Rotariu Eugen

Tiparul: Tiporex S.R.L.

Tiraj: 8.000 ex.

Preţ: 150 lei

Manuscrise originale sau listinguri de programe sînt primite cu plăcere de redacţie, cu condiţia să nu fi fost publicate şi în altă parte. Prin expedierea unui manuscris pe adresa redacţiei, autorul consimte implicit la publicarea materialului său în cadrul revistei. Onorariul se negociază cu redacţia. Materialele nepublicate nu se înapoiază şi nu se reţin.

Revista noastră vă oferă spaţiu pentru reclamă şi publicitate. Pentru amănunte vă rugăm să luaţi legătura cu redacţia.

Cei care doresc să anexeze revistei pliante publicitare tipărite în regie proprie, sînt rugaţi şi ei să se adreseze redacţiei.

Adresa şi telefonul redacţiei:
Micro ATCI, redacţia **"if"**,
RO-4300 Tîrgu Mureş,
C.P. 172, O.P. 1,
tel./fax 954/31660,
41417 sau 46885 (după ora 19)
telex 65354 (Intertours)

¤ **Peacock** - Monitoare Peacock Trinitron

Peacock Computer GmbH, Wünnenberg-Haaren, ofertant de soluţii profesionale PC complete, are disponibilă o nouă generaţie de monitoare. Aparatele se disting prin utilizarea unor tuburi cinescopice Black-Trinitron, care oferă o rezoluţie mai ridicată, contraste mai mari, împrospătarea mai rapidă a imaginii şi culori excelente. Sînt livrabile monitoare de 14, 17 şi 20 de inch, care în modul noninterlaced permit obţinerea tuturor rezoluţiilor VGA, pînă la o rezoluţie de 1280 X 1024 la monitoarele de 20 inch. Monitoarele color Peacock sînt realizate conform celor mai noi tehnologii în domeniu, iar radiaţiile electromagnetice emise nu depăşesc valorile stabilite de normele suedeze (MPR-II).

PEACOCK Computer GmbH,
Industriestr. 14, D-4798 Wünnenberg-Haaren,
tel. 02957/79-0

¤ **Dell** - Performanţele unui 486 la preţul unui 386

Firma Dell este prezentă pe piaţă cu o nouă serie de calculatoare interesante. Este vorba despre seria Dell 486P/50, P/33, P/25, P/20 şi P/16. Calculatorul Dell 486P/16 - 16 MHz i486SX, 4 MB RAM, 120 MB 17 ms harddisk, unităţi de dischetă de 3,5 şi 5,25 inch, monitor Super VGA, 14 inch, 1024 X 768 interlaced, inclusiv MS-DOS 5, Microsoft Windows 3 şi Microsoft mouse - costă doar 2.200$ în timp ce un 386/33 costă cca. 2.150$. De remarcat că viteza unităţii centrale a calculatorului Dell 486P/16 este cu 40% mai mare decît a unui Dell 333P la 33 MHz (test Landmark)!

O altă ofertă tentantă a firmei Dell este un Notebook color, Dell 325NC. Acest portabil color dispune de un procesor i386 SL, la 25 MHz, 64 KB cache, 4 MB RAM, expandabili pînă la 12 MB, 80 MB harddisk, lector de dischete de 3,5 inch, monitorul are diagonala de 9,25 inch şi poate afişa 16 culori în modul 640 X 480, şi pînă la 256 de culori în modul 320 X 200. Bateriile NiMH permit o funcţionare independentă de pînă la 3 ore. Preţ informativ: 4.000$.

System Plus International,
Bucureşti, str. Zăgazului 10, ap. 25,
tel/fax 90-791391.

¤ **Intel** - Soseşte 80586!

După cum a anunţat firma Intel, la mijlocul lui 1992 vor fi disponibile primele procesoare 80586. Acesta este un procesor pe 64 de biţi, care conţine peste 3 milioane de tranzistoare, deci cu 60% mai mult decît un 486. Intel afirmă că procesorul va fi de la patru pînă la cinci ori mai rapid decît un 486/33 MHz, şi va atinge 100 Dhrystone MIPS; pentru comparaţie amintim că un 486 atinge 41 Dhrystone MIPS. Structura internă a lui 586 corespunde cu tehnologia Superscalar RISC cu suport multiprocesor extins, astfel încît acest procesor va fi primul cip PC 100% compatibil RISC.

Intel GmbH, Dornacher Straße 1, 8016 Feldkirchen.

Apple în România

Romanian Computer Systems şi Delta Design au organizat în Bucureşti, la sala AGIR, în perioada 11-15 mai, o interesantă expoziţie Apple. Vă reamintim membrii familiei Apple:

Model	Procesor	RAM min./max.	Hard disk	Apple SuperDrive	Porturi SCSI/ADB	AppleTalk/ LocalTalk	Alte caracteristici
Macintosh Classic	8 MHz 68000	1 MB - 4 MB	40 MB	da	da	incorporat,	
Macintosh Classic II	16 MHz 68030	2 MB - 10 MB	40 sau 80 MB	da	da	incorporat	intrare/ieşire sunet
Macintosh LC	16 MHz 68020	2 MB - 10 MB	40 sau 80 MB	da	da	incorporat	emulare Apple IIe, intrare/ieşire sunet
Macintosh IIsi	20 MHz 68030	3 MB - 17 MB	40 sau 80 MB	da	da	incorporat	opţional NuBus sau 030 Direct Slot cu 68882
Macintosh IIci	25 MHz 68030/68882	5 MB - 32 MB	80 sau 160 MB	da	da	incorporat	trei sloturi NuBus, RAM cache
Macintosh IIfx	40 MHz 68030/68882	4 MB - 32 MB	80,160 sau 400 MB	1 - 2	da	incorporat	şase sloturi NuBus, RAM cache
Macintosh Quadra 700	25 MHz 68040	4 MB - 20 MB	80,160 sau 400 MB	da	da	incorporat	două sloturi NuBus, Ethernet, intrare/ieşire sunet
Macintosh Quadra 900	25 MHz 68040	4 MB - 64 MB	80,160 sau 400 MB	1 - 2	da	incorporat	cinci sloturi NuBus, Ethernet, intrare/ieşire sunet
PowerBook 100	16 MHz 68000	2 MB - 8 MB	20 MB intern	extern	da	incorporat,	
PowerBook 140	16 MHz 68030	2 MB - 8 MB	20 sau 40 MB intern	intern	da	incorporat	intrare/ieşire sunet
PowerBook 170	25 MHz 68030/68882	4 MB - 8 MB	40 MB	intern	da	incorporat	intrare/ieşire sunet, cartelă fax/modem

Lectorul de dischete SuperDrive poate citi şi scrie atît dischete format Macintosh cît şi MS-DOS. Bateriile modelelor portabile (ultimele 3) permit o funcţionare independentă pentru 2,5-4 ore. Merită menţionat şi faptul că pînă în toamna acestui an calculatoarele Apple vor "învăţa" limba română. Firma Apple este reprentată în România de firma Romanian Computer Systems, str. Luca Stroici, nr. 2, sector 2, Bucureşti.

IBM îşi extinde familiile de calculatoare PS/2 şi PS/1

Viena, 28 aprilie 1992 - IBM Eastern Europe îşi extinde familia sa de calculatoare PS/2 cu noi modele desktop şi notebook care utilizează puternicul procesor al IBM-ului 386SLC. IBM a anunţat de asemenea primele sisteme care vor utiliza noul microprocesor Intel 486DX2 şi adăugarea unor noi modele familiei de calculatoare PS/1.

"Odată cu introducerea acestor noi modele PS/2 şi PS/1 ne-am extins paleta de produse atît în familia sistemelor Micro Channel cît şi în cea a celor AT bus," a spus Dilip Chandra, director general IBM Eastern Europe. "Credem că aceste sisteme oferă raporturi preţ/performanţă excelente, la standardul de calitate IBM. Această tehnologie IBM oferă performanţe care nu pot fi depăşite pe piaţa calculatoarelor 386SX."

Noile sisteme AT bus la 20 MHz, Personal System/2 model 35 SLC şi 40 SLC caracterizează standardul 386SLC, oferind performanţe semnificativ îmbunătăţite faţă de standardul 386SX, multe aplicaţii rulînd cu pînă la 88% mai repede.

Puternicul cip 386SLC, proiectat şi fabricat în Burlington, Vermont, SUA, dispune de 8 KB cache intern şi de un cache controller intern. Performanţele acestuia cresc prin accedarea datelor din cache-ul intern, de viteză foarte ridicată, în loc de accedarea lor din memoria sistemului, ori de cîte ori acest lucru este posibil, şi prin optimizarea instrucţiunilor utilizate în mod curent. Ca rezultat, pe sistemele SLC rezultatele se vor observa mai rapid decît pe majoritatea sistemelor 386DX-25MHz.

PS/2 Model 35 SLC se livrează cu harddisk-uri de 40 MB sau de 80 MB, iar modelul 40 SLC cu disc de 80 MB. Toate trei modelele sînt livrate în mod standard cu un lector de dischete de 2,88 MB.

Sînt anunţate noi modele PS/1

IBM a anunţat, de asemenea, şi modelul PS/1 Pro 386 SX care extinde familia de produse PS/1 cu sisteme mai puternice şi cu mai multe facilităţi de extensie, care să satisfacă necesităţile utilizatorilor atît pentru munca la domiciliu cît şi la birou.

Toate modelele PS/1 Pro includ un procesor 386SX/20 MHz, 2 MB memorie extensibilă pînă la 16 MB, trei slot-uri de extensie AT bus, grafică VGA pe 16 biţi şi monitor color 14 inch incorporate, tastatură cu 102 taste şi mouse. Pe harddisk sînt livrate soft- urile: IBM DOS 5.0, Microsoft Works 2.0 şi Microsoft Windows 3.0. PS/1 Pro este disponibil în trei variante: cu harrdisk de 40 MB, 80 MB sau 129 MB, şi capacităţi de extensie pentru un al doilea disc.

Un nou notebook SLC

IBM îşi extinde şi producţia de modele notebook, odată cu anunţarea sistemului PS/2 Model N51 SLC, primul portabil care profită de tehnologia SLC. Dezvoltat şi fabricat de IBM, procesorul SLC/16 MHz este de pînă la 75% mai rapid decît un procesor standard 386SX/16 MHz, oferind o putere de calcul echivalentă cu cea a unui procesor 386SL/25 MHz.

PS/2 Model N51 SLC dispune de 2 MB RAM, expandabil pînă la 10 MB, un harddisk de 80 MB, un lector de dischete de 2,88 MB, un monitor LCD (Liquid Crystal Display) VGA cu 32 de nuanţe de gri, şi două baterii, nichel metal hidrid, o tastatură cu 85 de taste, şi un port de extensie Micro Channel.

Capacitatea mare de stocare şi puterea procesorului recomandă utilizarea modelului N51 SLC sub sisteme de operare avansate, cum ar fi OS/2 sau pentru rularea unor aplicaţii care să beneficieze de viteza procesorului, cum ar fi aplicaţiile numerice sau grafice.

Sînt disponibile noi sisteme 486DX2

IBM a anunţat, de asemenea, două noi modele 95 tower şi două modele 90 desktop, care utilizează procesorul Intel 486DX2. Acest nou cip utilizează o tehnologie de "dublare a vitezei" care permite ca intern procesorul să lucreze la 50 MHz, în timp ce extern să se lucreze doar la 25 MHz. Rezultatul este un sistem cu performanţe cu pînă la 75% mai ridicate în mediul staţiilor de lucru, şi cu performanţe cu pînă la 18% mai ridicate în domeniul fileserver-elor, comparate cu un procesor standard 486SX/25 MHz.

ROMORGADATA '92

Romexpo S.A. şi IEG Solingen au organizat în perioada 5-8 mai 1992, la Complexul expozitional din Bucureşti, expoziţiile specializate ROMMEDICA, ROMPHARMA, ROMCONTROLA şi ROMORGADATA. Deşi s-a pus accentul în primul rînd pe aparatura medicală, farmacie şi tehnică de măsură şi control, firme mari (şi) în tehnica de calcul avînd standuri numai de interes medical (Hewlett-Packard, Toshiba, Philips), şi-a avut şi secţiunea de tehnică de calcul şi informatică ponderea ei, printre expozanţi numărîndu-se: Adisan, B.F.D. Import-Export SRL, Electric Business SRL, ICI, Intercom SRL, Kontrax Birotica SRL, KT Technology Ltd., DHS Merger & Partner GmbH, Misaco SRL, Rom Team Solutions SRL, Scop, SINTA SA, SIS, System Plus SRL, Tektronix.

CERF '92

În perioada 11-15 mai, 1992, la Sala Polivalentă din Bucureşti a avut loc expoziţia de calculatoare şi produse electronice CERF '92 (Computers and Electronics Romanian Faire), expoziţie organizată de Comtek Expositions, Inc. SUA şi sponsorizată de firma Computerland Bucureşti. Expoziţia, o primă încercare de rafinare a profilului, a reunit nume de marcă din domeniul tehnicii de calcul şi informaticii: Alfa Bit SRL, Computerland Bucureşti, DHS Merger & Partner GmbH, Trend SRL, ICE Felix Computers, Innovator SRL, Laser Computer Romania SA, Lecom SRL, Misaco, Rom Control Data, Romtrust Inc., Style Impex SRL, Sysgraph Computergraphik GmbH, Unisys, System Plus International, Westech Co. SRL, Universys SRL, Rom Team Solutions.

THE deal

in p.c. field

HARDWARE

SOFTWARE

CAD - CAM

DESKTOP PUBLISHING

Technical support

TRAINING

Consulting

雅仕 A&C INTERNATIONAL S.A.

Tel 40-0-53.53.15.

Puterea unei workstation la nivel DOS

Washington, D.C., 31 martie 1992: Quarterdeck Office Systems, Inc. (NASDAQ: QDEK) a făcut cunoscut, în cadrul expoziţiei Federal Office Systems Expo (FOSE), faptul că s-a început livrarea produsului DESQview/X. Produsul permite utilizarea unora dintre facilităţilor existente pînă acum numai pe staţiile de lucru (workstations) atît pe DOS-PC-uri, cît şi pe PC-uri 386 individuale sau cuplate în reţea. DESQview/X, al treilea produs Quarterdeck din seria de produse multitasking, este un program multitasking şi multifereastră grafic. Printre caracteristicile sale se numără, printre altele, un desktop grafic, fonturi scalabile, macrouri de tastatură şi meniuri adaptabile, transfer de date şi remote computing. O caracteristică importantă a lui DESKview/X este facilitatea de a rula atît programe DOS în mod text, cît şi programe grafice Microsoft Windows în ferestre mici sau "remote" pe un alt PC DESKwiew/X sau pe o X-Workstation.

DESKwiew/X este prima implementare client/server completă la nivel DOS a sistemului X-Window - un standard internaţional pentru sisteme grafice cu tehnica ferestrelor şi partajarea puterii de calcul. Pentru a putea atinge performanţele unei workstation pe lîngă facilitatea de a rula simultan programe DOS şi Microsoft Windows (multitasking), DESKwiew/X conţine şi sistemul grafic şi protocoalele de reţea X-Window şi, suplimentar, tehnologia "shared" DOS-Extender pe 16 şi 32 de biţi, a firmei Rational Systems.

Avantajele aduse de DESQview/X pentru PC-uri individuale sau cuplate în reţea:

- transfer de date - cut şi paste pentru aplicaţii care rulează în diferite ferestre, indiferent dacă acestea lucrează REMOTE (la distanţă) sau LOCAL;
- suprafaţă utilizator grafică cu "look and feel" tridimensional;
- meniuri şi icon-uri uşor adaptabile;
- sistem avansat de gestiune a fişierelor şi imprimantelor;

La livrare DESQview/X mai este însoţit de:

- DESQview/X Application Manager, un desktop grafic şi un manager de programe;
- DESQview/X File Manager, un sistem de gestiune a fişierelor locale sau remote, DOS sau neDOS;
- DESQview/X Icon Editor, un instrument grafic pentru crearea şi editarea de icon-uri pixmap şi bitmap;
- Adobe Type Manager (ATM) for DESQview/X;
  - un Outline Font Manager cu 13 fonturi Typ 1 pentru DESQview/X şi programele grafice pentru sistemele X-Window;
  - permite şi ferestre DOS scalabile pentru programe DOS în mod text.

După cum afirmă Gary Pope, vicepreşedinte în domeniul proiectării şi unul dintre membrii fondatori ai firmei Quarterdeck, pînă acum la nivel DOS lipsea un sistem integrat de ferestre conceput pentru comunicarea grafică. Prin portarea sistemului X-Window la nivel DOS, utilizatorii dintr-o reţea pot accede la întregul X- Software, care rulează pe workstations Unix sau VAX sau pe calculatoare mainframe.

Împreună cu DESQview/X se mai livrează bestseller-ul Quarterdeck QEMM-386 (un manager de memorie), Manifest, un instrument de analiză a sistemului şi memoriei şi un manager de reţea pentru reţele Novell Netware şi NetBIOS (575 DM). Opţional se mai livrează un manager de reţea pentru TCP/IP, DESQview/X OPEN LOOK Window Manager şi DESQview/X OSF/Motif Window Manager.

Cerinţe hard: PC 386 compatibil IBM sau mai mare; PC sau MS-DOS 3.3 sau mai mare sau DR DOS 6; 4 MB RAM sau mai mult, cartelă grafică EGA, VGA, SVGA sau DGIS cu 256 de culori; harddisk de cel puţin 40 MB. În prezent se lucrează şi la o versiune pentru PC- uri 286.

Quarterdeck Office Systems GmbH, Willstätter Straße 15, D-4000 Düsseldorf 11, tel. (49) 211-59790-0, fax: (49)211-594126.

PACO SOFT

O surpriză plăcută la ROMORGADATA a constituit-o standul "Diviziei de informatică" a Palatului Copiilor. Sponsorizat de firmele Scop, Megalan şi Logic, Palatul Copiilor a reuşit ca de la resursele materiale şi umane existente în septembrie 1990 (50 X HC 85, 2 X HC 85 extins, 2 X TIM S, soft Spectrum-Sinclair, 3 profesori de informatică) să crească pînă în septembrie 1991, resurselor hard iniţiale adăugîndu-li-se o reţea Novell (11 PC- uri), o imprimantă laser HP Laserjet IIP şi una matricială Epson FX 1050, iar celor soft adăugîndu-li-se produsele firmei Borland. Divizia, care este o asociaţie care nu-şi propune realizarea de profit, lucrează cu cca. 1500 de copii, în diverse stadii de pregătire: Basic începători, Basic avansaţi, C începători, C avansaţi, reţele de calculatoare, cei 5 profesori de informatică şi 8 colaboratori fiind insuficienţi pentru acoperirea necesarului. În cadrul expoziţiei au fost prezentate mai multe aplicaţii în sfera reţelelor locale de calculatoare, programele (sursă şi executabile) fiind distribuite gratuit celor care aveau la îndemînă o dischetă goală, după terminarea expoziţiei aceste programe putînd fi obţinute de la firmele sponsorizatoare. Enumerăm cîteva dintre aceste programe:

PACOmail, pachet de poştă electronică, autor: Şerbu Ciprian, clasa a VIII-a

PACOchat, utilitar care permite comunicarea între staţii în regim conversaţional, autor: Gâlcă Mihnea, clasa a VIII-a

PACOprint, print server, autor: Fulger Dan, clasa a IV-a

PACOdir, utilitar care permite unui utilizator să afle toate directoarele în care se află fişiere care îi aparţin, autor: Romaşcanu Vlad, clasa a VII-a

PACOprof, pachet de utilizare dualã (profesor/elev) destinat procesului de instruire. Autor: Damian Ionuţ, clasa a IX-a.

PACOcls, pachet dual de tip compile & link server care permite unui utilizator de la o staţie mai puţin dotată (de ex., un XT) să lanseze o compilare şi/sau o linkeditare dificilă pe o staţie mai dotată (de ex., un 386 cu 4MB RAM), autori: Eşeanu Costin, clasa a VIII-a şi Băloiu Liviu, clasa a VII-a

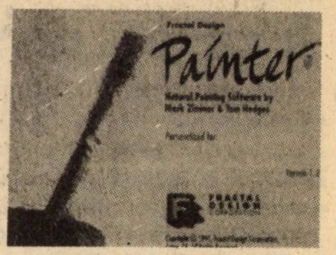
PACOgenf, generator de propoziţii pornind de la o matrice de cuvinte, autor: Popa Sabin, clasa a IV-a

Palatul Copiilor - Divizia de informatică
B-dul Tineretului nr. 8-10, sector 4
tel. 754900, 754083, fax 757135, 150838

Fractal Design Painter 1.2b2

A picta ca van Gogh

Pînă acum a existat o diferenţă imensă între pictura pe pînză şi cea pe ecran. Cu programul Painter al Corporaţiei Fractal Design acest lucru se va schimba radical: programul simulează tehnicile naturale de pictare pe baza algoritmilor de fractali.



Painter for Windows

Prima dată Painter a fost prezentat la expoziţia de toamnă Comdex de la Las Vegas, de atunci a fost găsit şi un distribuitor german, şi anume Letraset şi apariţia versiunii Windows a fost anunţată pentru următoarele săptămîni.

Suprafaţa şi modul de descriere

Imediat după pornirea programului, acesta se prezintă printr-o suprafaţă care aminteşte de outfit-ul versiunii Macintosh: toate sculele şi toţi parametrii sînt grupaţi în toolbox-uri grafice foarte perfecţionate, care nu posedă capetele de tabel tipice Windows, ci provin direct de la suprafaţa Mac. Aceste toolbox-uri pot fi dispuse liber pe ecran sau ele pot fi închise provizoriu.

Painter se deserveşte fie cu un mouse fie cu o tabletă grafică. În "pachetul" de livrare se află pe lîngă un driver special de mouse, care ajută la o transmitere mai precisă a mişcărilor mouse-ului, şi drivere pentru digitizorul SD-510 şi SD-42x al firmei Wacom şi pentru tableta de desen Calcomps Drawingboard II. Primele încercări cu SD-510 arată că doar în legătură cu acest aparat de intrare programul îşi poate dezvolta întreaga capacitate creatoare. Utilizatorul lucrează cu un creion fără cablu şi fără baterie, care permite pe de o parte o conducere mai naturală a pensulei decît cu un mouse. Pe de altă parte acesta (creionul) este sensibil la presiune, grosimea culorii se schimbă în funcţie de apăsarea, care se aplică de utilizator creionului.

Scule de pictat, hîrtie şi culori

Toate funcţiile oferite de acest software folosesc simulării exacte a sculelor şi tehnicilor naturale de pictat.

Elementul central este o colecţie de scule, care conţine printre altele: pensulă de ulei, marker de text, vîrfuri de cariocă şi ceară, cretă, cărbune şi chiar un airbrush. Există chiar şi funcţii care simulează stilul lui van Gogh, respectiv al lui Seurat. Toate sculele pot fi manevrate, avînd în vedere modul lor de funcţionare, prin diferiţi parametri şi în cele mai multe cazuri ele reflectă o imagine foarte precisă a realităţii. De exemplu, se poate observa că un marker de text galben devine cu timpul tot mai murdar, dacă cu el se colorează peste culori mai închise. Painter face diferenţă între acuarele umede şi uscate, el cunoaşte procedeul de amestec şi de acoperire a diferitelor tipuri de culoare şi ţine cont de structura hîrtiei folosite la aplicarea culorii.

Felul hîrtiei se alege în altă boxă de dialog. Utilizatorului îi stau la dispoziţie diferite sortimente de pînză şi hîrtie, dar se pot alege şi materiale textile, tapete din fibră dură şi alte materiale. Aceste setări se folosesc apoi de către Painter la folosirea diferitelor scule, cum ar fi cărbune sau vîrf de ceară.

O imagine terminată poate fi acoperită complet cu o textură dorită de hîrtie cu ajutorul unei funcţii speciale, iar programul mai are în vedere şi direcţia de incidenţă a luminii asupra tabloului.

Există şi o dependenţă directă între sculele de pictat folosite şi paleta de culori Painter. Cu toate că oricînd poate fi aleasă orice culoare din paleta de culori, în plus programul mai oferă şi o altă paletă, din care se pot alege cîte 15 culori tipice pentru o sculă de pictat.

Alte funcţiuni

Painter nu se pretează doar pentru artişti, ci oferă şi pictorului mai puţin versat unele funcţii interesante. Există de exemplu posibilitate de a încărca fişiere bitmap şi de a le pune sub un fel de hîrtie pergament. Dacă apoi se pictează cu o aşa-numită sculă de clonare deasupra acestei hîrtii de tracing, Painter combină funcţionalitatea sculei cu coloristica imaginii încărcate, se dă astfel posibilitatea de a copia o imagine.

Se păstrează în acest timp întregul control asupra mişcării pensulei, şi nu este nevoie să urmărim exact culoarea imaginii originale.

Auto-Cloning-ul merge cu un pas mai departe. Această funcţie amortizează procesul descris mai sus, el preia deci şi pictatul propriu-zis. Urmează ca în final imaginea să fie pusă pe o textură de hîrtie corespunzătoare, şi deja tabloul generat de calculator este terminat.

Mai există bineînţeles şi alte funcţii ajutătoare, cum ar fi o rutină de umplere, posibilitate de a genera şabloane şi o lupă pentru lucrări mai precise.

În schimb, s-a renunţat la funcţia text - aceasta nu a avut-o la dispoziţie nici chiar van Gogh.

Concluzie

Painter este unul dintre programele de grafică cele mai inovative, elaborate în ultimii ani. Astfel, graficianului ce utilizează un PC îi stă la dispoziţie un program autentic de pictare, după ce Deluxe Paint a murit de mult, iar produsele Rix şi ZSoft s-au specializat în prelucrarea de imagini. Singura problemă: cel ce doreşte să se bucure de Painter, acela are nevoie de un calculator destul de rapid cu o memorie de lucru disponibilă de cel puţin 6 MByte şi de o cartelă SuperVGA.

Letraset Deutschland GmbH
Mergenthaler Straße 6
6000 Frankfurt 60
Tel. 069/4209940

Alpha - o nouă tehnologie a procesoarelor

Ultimul chip RISC prezentat de Digital Equipment nu poate fi întrecut încă din punct de vedere al vitezei, al scalabilităţii şi a multitudinii sistemelor de operare care îl pot utiliza.

Digital Equipment a dezvăluit sub denumirea seacă de "21064" secretul păzit cu străşnicie timp de 12 ani de cercetare şi dezvoltare: un procesor RISC de 64 de biţi, care, la o frecvenţă de tact de 150 MHz, dezvoltă o putere maximă de 300 MIPS. Cei din branşă, mai ales cei de la firmele Mips şi DEC, ambele membre ACE, au rămas surprinşi. Nu de multă vreme cei din consorţiul ACE s-au pus de acord asupra unui singur procesor "R4000" al firmei Mips, pentru sistemele de operare ACE, "Windows NT" şi "Open Desktop". Din punct de vedere al performanţelor şi perspectivelor de viitor, CPU-ul firmei DEC este cu mult înaintea lui Mips.

Procesorul 21064, denumit pe scurt "Alpha" reprezintă prima implementare a unei arhitecturi care se poate afirma cîţiva ani buni datorită unei adaptabilităţi crescute. Implementări ulterioare pot fi executate la frecvenţe da tact mai ridicate. Din auzite, există deja un model de laborator care funcţionează la 300 MHz.

Pentru fabricare lui Alpha, care s-a realizat printr-o tehnologie de 0,75 micrometri-CMOS-4 cu patru planuri de metalizare, a fost necesară construirea unei fabrici de 460 milioane de mărci în Scoţia. Costurile de dezvoltare probabil au fost şi mai ridicate.

Procesorul veritabil de 64 de biţi, cu regiştri de 64 de biţi, cu bus-uri de date de 64 de biţi, cu adrese de 64 de biţi şi cu operaţii pe 64 de biţi este primul model al unei noi familii de procesoare. În timp ce alte CPU-uri, care lucrează intern tot cu 100 MHz, reprezintă apusul unei familii, modelul firmei DEC la o frecvenţă de tact de 150 MHz are foarte bune perspective de viitor.

S-ar putea discuta despre paleta de aplicaţii posibilă, începînd de la Palmtop pînă la Supercomputere, deoarece la o putere absorbită între 20 şi 30 de waţi orice producător de motherboard-uri pentru calculatoare Desktop ar fi tentat să-l utilizeze, nemaivorbind despre producătorul unui Palmtop. Pe o suprafaţă de 17x14 milimetri se găsesc 1,7 milioane de tranzistoare. Dintre acestea memoria Cache şi alte memorii ocupă cca. 840.000 de tranzistoare. Pentru asigurarea logicii au mai fost necesare aproximativ 800.000 de tranzistoare. Partea considerabilă de logică se opune unei miniaturizări mai mari a cipului.

Procesorul este astfel conceput încît programele Unix şi OSF/1 pot fi folosite în acelaşi fel ca şi software-ul VMS propriu.

Arhitectura are în vedere atît sisteme multiprocesor cît şi sisteme de calculatoare paralele, care se bucură de o tot mai mare răspîndire.

Alpha este o arhitectură tipică Load/Store-RISC. Înseamnă că toate operaţiile, cu excepţia operaţiilor Load/Store, se efectuează între regiştri. 32 de regiştri Integer şi 32 de regiştri Floating-Point au fiecare lăţimea de 64 de biţi.

Toate comenzile au o lăţime de 32 de biţi tipice tuturor arhitecturilor RISC, avînd patru formate diferite şi un cod operand de 6 biţi.

O caracteristică care deosebeşte arhitectura Alpha de altele este instrucţiunea denumită PAL-Call. Este vorba de fapt de cîteva zeci de comenzi complexe care tratează întreruperile, stările de excepţie, adresările virtuale sau task switching. Aceste subprograme pot fi adaptate foarte precis la un anumit sistem de operare. Se poate astfel rula VMS dacă se alege acea versiune PAL-Call, care este adaptată multora dintre posibilităţile sistemului de operare VAX. Acelaşi lucru este valabil şi pentru OSF/1 sau Windows NT. Astfel Alpha devine interesant pentru multe sisteme de operare.

Pentru procesoarele care se orientează după arhitectura procesoarelor RISC, compilatorul folosit este de mare importanţă, deoarece acesta trebuie să producă întotdeauna un şir de comenzi maşină prelucrabile.

DEC a dezvoltat un compilator care se compune dintr-o parte Front-End şi una Back-End. Cu aceasta ar fi susţinute chiar şi sisteme tolerante la erori şi platforme în timp real.

Cele mai importante blocuri de funcţii ale procesorului Alpha sînt o unitate Integer de 64 de biţi şi o unitate Floating-Point tot de 64 biţi. Prima dintre ele lucrează cu un Pipeline de 7 trepte iar cea de-a 2-a, chiar cu una de 10 trepte. Cipul este completat de o unitate Load/Store sau de adrese, care răspunde de handling-ul ramificaţiilor programului şi de salturi.

Datorită arhitecturii Superpipeline în mai multe trepte se asigură pe ciclu cîte o instrucţiune fiecărei funcţii. Deoarece în paralel fiecare Pipeline emite cîte o comandă/tact în total se execută 2 comenzi pe un tact. Doar înmulţirile Integer şi împărţirile Floating Point nu sînt total sprijinite de Pipeline. Dintre cele patru tipuri de comenzi de bază întotdeauna se poate executa o pereche de comenzi: Load/Store, operaţii Integer, operaţii Floating-Point şi operaţii de ramnificare/salt. Comenzile Load/Stare pot prelucra atît 32 cît şi 64 biţi.

Unitatea Floating-Point susţine atît tipuri de date standard VAX cît şi cele ale standardului IEEE. Cu excepţia împărţirii unitatea predă pe ciclu un rezultat pe 64 de biţi.

Pe cip sînt integrate două memorii cache pentru date şi pentru instrucţiuni, de cîte 8 KByte. Memoria cache pentru date este executată ca un cache "Write through" şi "Direct mapped" cu blocuri de 32 biţi. Şi cache-ul de comenzi este "Direct mapped" şi are aceeaşi dimensiune. Se susţine un cache secundar printr-o interfaţă programabilă pînă la o mărime de 8 MByte.

Generatorul de tact integrat este responsabil de tactul relativ înalt al procesorului, dar şi de frecvenţele de tact programabile CPU tact/2 şi CPU tact/8.

Lăţimea busului de date poate fi aleasă între 64 şi 128 biţi la frecvenţe de tact cuprinse între 75 şi 18,5 MHz. Un numărător ON-CHIP măsoară mereu puterea CPU-ului şi a sistemului. Spaţiul fizic al adreselor măsoară în prezent 34 biţi, cel virtual doar 43 biţi, dar ambele au fost încercate la 64 biţi. Motivul acestei restricţii este numărul de conexiuni al cipului (de pini) de "doar" 431. Din acest punct de vedere următoarele versiuni vor oferi mai multe posibilităţi.

Dar şi cu actualul mod de realizare memoria adresabilă de cîteva mii de GBytes este astronomic de mare şi ea se dublează cu fiecare pin de adresă. Creşterea puterii de calcul nu se va mai putea exprima pe viitor doar prin frecvenţa de tact, trebuie avut în vedere, şi numărul de instrucţiuni pe care le poate executa procesorul simultan. Procesorul 21064 poate executa 2 instrucţiuni pe ciclu, şi se preconizează o rată de 8 instrucţiuni pe ciclu.

Primele chipuri de acest tip vor fi livrate în următoarele cîteva luni, dar cantităţi notabile pot fi aşteptate doar la începutul anului 1993. El va costa 3.400$ bucata. Angrosiştii îl pot cumpăra la un preţ de aproximativ 1.500$ bucata.

THE right option

MIX® COMPUTERS

with

ATI Graphic Cards

MAG Monitors

PC MAGAZINE EDITOR'S CHOICE

1991 BYTE AWARD OF EXCELENCE

CalComp

Panasonic

HEWLETT PACKARD

in hardware

雅仕 A&C INTERNATIONAL S.A.

Tel 40-0-53.53.15.

Head Up Display

Tehnologia de proiecţie pentru afişajele militare montate pe cască (HMTV - Helmet Mounted TV) a evoluat continuu de-a lungul a peste 30 de ani, dar numai recent a fost adaptată cu succes la utilizări civile.

Acum, sistemele personale de afişaj prin proiecţie, miniaturizate, ieftine, îşi găsesc aplicaţii într-o gamă de utilizări comerciale şi industriale în pînă expansiune.

Ecranele virtuale sînt sisteme de proiecţie a imaginii ce permit utilizatorului să priveasscă într-un dispozitiv mic, situat aproape de ochi şi să vadă o imagine (a informaţiilor de afişat) de dimensiune mult mai mare şi situată la o distanţă aparentă mai mare. Cu respectivul dispozitiv de mici dimensiuni montat pe cască sau pe "ochelari", afişajul se află permanent în cîmpul vizual personal al utilizatorului, care astfel este liber să efectueze alte sarcini.

Ecranul virtual montat pe cască, bazat pe un tub catodic (CRT) în miniatură pentru aplicaţii militare critice, a asigurat HUD ("afişaj cu capul sus") ce permite piloţilor şi conducătorilor de tancuri să supravegheze instrumentele indicatoare şi sistemele de achiziţie a ţintelor, fără a întrerupe contactul vizual direct cu scenele reale de pe cîmpul de luptă.

Ce este HUD?

Head Up Display, "afişaj cu capul sus". Este un dispozitiv utilizat iniţial la avioanele militare, dar întîlnit acum din ce în ce mai des în cokpit-ul avioanelor civile.

Permite afişarea prin proiecţie a informaţiilor vitale pentru zbor, care se găsesc astfel regrupate în faţa ochilor pilotului, ceea ce evită aplecarea în permanenţă a privirii pentru consultarea tabloului cu instrumentele de bord.

La avioanle de luptă, se mai proiectează în plus diferite imagini reale sau sintetice (hartă electronică) ale peisajului survolat sau ale ţintelor vizate (în infraroşu, radar).

Imaginea este proiectată deasupra bordului, în conturul cîmpului vizual al pilotului, iar distanţa aparentă de focalizare pentru ochi este infinită, astfel că nu solicită modificări ale acomodării ochilor (cum este cazul cînd se schimbă privirea de la mediul ambiant la panoul de bord)

Primul ecran virtual practic, bazat pe tub catodic, a fost introdus de firma Hughes Aircraft în 1958. Aproape toate performanţele ulterioare s-au bazat doar pe îmbunătăţirea afişajelor miniatură cu tub catodic, şi-n consecinţă au fost limitate de caracteristicile lor inerente de cost, rezoluţie şi dimensiuni.

Au existat şi sisteme prototip, nebazate pe tub catodic, cele mai multe cu cristale lichide microminiaturâ şi sisteme optice de mărire. Dar acestea sînt dificl şi costisitor de fabricat, iar performanţele lor n-au fost la fel de bune ca la cele cu tub catodic.

Un nou tip de dispozitiv de afişare prin proiecţie.

În 1990, o firmă din New England a introdus un nou tip de dispozitiv, proiectat special pentru piaţa civilă, nu militară, şi bazat pe o tehnică diferită. Denumit Private Eye, este fabricat de Reflection Technology Inc., din Waltham, Massachussets şi promite să înlocuiască afişajele cu tub catodic în toate aplicaţiile de ecran virtual monocrom, cu excepţia celor foarte critice.

În timp ce sistemle precedente utilizau un afişaj bidimensional şi un sistem optic fix, noul dispozitiv foloseşte o reţea unidimensională (baretă de LED-uri) pentru afişaj şi o optică cu baleiere.

Principial, dispozitivul se compune dintr-o baretă de LED-uri (diode electroluminescente) cu electronica de comandă aferentă, un obiectiv de proiecţie şi o oglindă mobilă, montată pe arcuri, ce reflectă lumina spre ochiul utilizatorului.

Imaginea de afişat este codată digital şi trimisă la electronica de comandă printr-o interfaţă serie, sub forma unei hărţi de biţi (bit map). Harta de biţi este stocată în memoria de ecran şi organizată în format de coloane de către o reţea de porţi la comandă (circuit specializat, cu reţea de porţi logice interconectate discreţionar, ce realizează algoritmul dorit).

Mecanismul de bază al afişajului constă dintr-o reţea liniară, monolitică de LED-uri - 280 în linie - un obiectiv ce măreşte imaginea şi o oglindă oscilantă, cu contragreutate mobilă. Oglinda este comandată de un motor miniatură cu două înfăşurări. Bobinele sînt controlate de un sistem de servoreglaj în buclă închisă. Oglinda, montată pe arcuri, este echilibrată cu o contragreutate mobilă, pentru a absorbi cea mai mare parte a vibraţiilor mecanice parazite.

În fiecare moment utilizatorul, privind în oglindă, vede proiecţia reţelei verticale de LED-uri, într-o poziţie orizontală corespunzătoare unei coloane de date de afişat. Oglinda este balansată pe orizontală cu o frecvenţă fixă, de rezonanţă, baleind un arc de 15 grade.

Controlerul de afişaj trimite datele de coloană unei perechi de cipuri de registre de decalaj CMOS, fiecare comandînd 140 de LED-uri. LED-urile, în sincronism cu mişcarea oglinzii oscilante, afişează coloana de date de imagine corespunzătoare poziţiei instantanee a oglinzii.

Pentru a sincroniza bareta de LED-uri cu poziţia oglinzii sînt montate o sursă de lumină staţionară şi un fotoreceptor. O plăcuţă (tab) de pe spatele oglinzii permite luminii de referinţă să ajungă la senzor atunci cînd oglinda se apropie de deflexia maximă, generînd astfel un puls de sincronizare (semnal de sfîrşit de cursă/linie).

Rata de baleiere pe orizontală a oglinzii, de 50 Hz, este destul de ridicată pentru ca utilizatorul să vadă, fără pîlpîire, o imagine virtuală a întregului afişaj bidimensional, cu o rezoluţie de 720 (H) x 280 (V) pixeli.

În cazul afişării de texte, dispozitivul asigură afişarea a 25 de rînduri normale, a cîte 80 de caractere. Imaginea virtuală, roşie pe fond negru, are dimensiunea aparentă de 12" (31 cm) şi apare la circa 2 ft (0,66 m) de ochiul utilizatorului.

Un reglaj surpinzător de important pentru o vizionare subiectivă optimă este ajustarea fină a claritâţii imaginii. De fapt, nu-i vorba propriu-zis de claritate, ochiul adaptîndu-se automat, ci de distanţa aparentă de proiecţie. Se realizează printr-un reglaj culisant al focalizării: ansamblul lentilelor-obiectiv glisează pe o şină, şi poate fi reglat de

utilizator astfel ca imaginea proiectată să apară oriunde între 9" (23 cm) şi infinit, faţă de ochiul utilizatorului.

Astfel, imaginea afişajului poate fi plasată la aceaşi distanţă cu obiectele fizice (reale) din cîmpul vizual al utilizatorului, evitînd oboseala ochilor datorată permanentelor schimbări de focalizare (reglare a distanţei de punere la punct).

Considerente de încapsulare

Încapsularea este un factor esenţial în simplificarea producţiei şi reducerea costurilor. Un singur circuit integrat hibrid încorporează reţeaua (bareta) de LED-uri şi driver-ele de curent asociate. Acest circuit hibrid, împreună cu controller-ul buclei servo a motorului, memoria de ecran şi totalitatea componentelor discrete, este asamblat în tehnologia montării pe suprafaţă pe o placă de cablaj flexibil, ce se pliază într-o cutie mică de plastic. Tot dispozitivul măsoară 2,5x2,5x7,6 cm şi cîntăreşte circa 70 g. Consumul de energie este de 0,33W la +5V. Dispozitivul poate fi produs în serie mare, la un preţ de cost mult mai mic decît cel al ecranelor virtuale cu tub catodic miniatură.

Dimensiunile şi greutatea reduse ale corpului deschid perspective pentru numeroase aplicaţii potenţiale. Astfel, fabricanţii de calculatoare portabile (laptop) intenţionează să ofere modele noi, bazate pe o variantă de "Private Eye". Se înlocuieşte afişajul clasic (cristale lichide sau panou cu plasmă), real, cu un afişaj virtual montat pe headset (bentiţă, inel de susţinere pe frunte, ca cel pentru oglinda oftalmologică) la cîţiva cm de ochi.

Aceasta permite operatorului aflat în tranzit să lucreze mai bine într-un spaţiu restrîns, mai uşor şi mai confortabil. Imaginea de pe ecranul virtual este permanent în cîmpul vizual personal (şi privat - nu-l vede altcineva) al operatorului.

În domeniul servis/mentenanţă/depanare echipamente, afişajul virtual este deja folosit ca o parte a manualului de servis cu diagnostic asistat de calculator, pentru a optimiza eficienţa (pleonasm?) reparaţiilor.

Aici, pagina din manual referitoare la subansamblul în curs de inspectare sau depanare este proiectată în faţa tehnicianului care lucrează. În consecinţă, tehnicianul poate lucra cu ochii numai pe echipament şi cu ambele mîini libere, minimizîndu-se astfel riscul de accidente şi erorile costisitoare. Aceasta este o capabilitate importantă şi promiţătoare, în special atunci cînd munca de depanare trebuie efectuată într-un mediu cu grad ridicat de periculozitate sau precar (?).

În domeniul depanării asistate (dar nu numai), Reflective Technology intenţionează să folosească discurile laser (Compact Disk) drept mediul prin intermediul căruia volumele arhivate de manuale de servis şi "cărţi de electronică" vor fi redate pe ecranul virtual miniatură.

Notă: Cu vreo 10 ani în urmă se prezenta la Teleenciclopedia "cartea electronică". Informaţiile erau stocate într-o memorie moartă (ROM, EPROM) şi afişate pe un ecran cu cristale lichide. Eu făcusem socoteala că densitatea de informaţie este mai mică decît la o carte clasică, şi-i evident mult mai scumpă. Pe atunci preţul memoriilor era destul de ridicat, iar capacitatea maximă, de 64 K. În schimb, varianta cu CD-ROM şi ecran virtual "Private Eye" pare promiţătoare, atît sub aspectul capacităţii (200.000 pagini pe disc de 5") cît şi al costului (sub 3$ pentru CD-urile audio sau video).

Fax fără hîrtie

O firmă comercializează curent un receptor facsimil fără hîrtie, folosind o variantă a lui "Private Eye". Acest echipament poate stoca pînă la 25 de pagini de date transmise, care pot fi apoi parcurse (scroll) de operator şi vizualizate pe un ecran virtual cu beneficiul suplimentar al unei depline intimităţi (nu mai poate să ţi se uite cineva peste umăr pe ecran).

Notă: Aceasta nu înseamnă că în curînd toate faxurile vor fi fără hîrtie. Noţiunea de "imprimantă fără hîrtie" nu are sens. Sînt destule situaţii în care ai nevoie de o copie permanentă a imaginii de pe ecran, unde capabilitatea de hard-copy este importantă. Dar există, desigur, domenii în care nu-i necesară o înregistrare permanentă, informarea trebuie să fie rapidă şi confidenţială, scop în care faxul cu ecran virtual (eventual racordat la reţeaua telefonică fără fir) este indispensabil.

"Reflective Technology" oferă de asemenea o placă - driver care lucrează cu calculatoarele personale sub MS-DOS şi asigură interfaţarea acestor echipamente cu produsele tip ecran virtual ale firmei RT.

În următorii 5 ani, ţelul firmei de a lansa pe piaţă un afişor cu scală de griuri, în culori, cu rezoluţia de 1024x1024 (1M pixeli), într-o capsulă de dimensiuni mai mici decît cea curentă. Această versiune viitoare va avea grafică complet automată, la un preţ suficient de scăzut pentru aplicaţii electronice de larg consum - esenţialmente un televizor de înaltă definiţie (HDTV) într-o cutie de chibrituri.

Notă: Da, se poate considera un breakthrough (străpungere tehnologică, pătrundere revoluţionară). Pînă acum, afişajul - ecranul - constituia un obstacol major în calea miniaturizării receptoarelor TV. Ce folos că toate funcţiile aparatului sînt îndeplinite de un singur cip, dacă ecranul este mare şi consumă mult?

Iar variantele comerciale de ecrane cu cristale lichide color, precum Watchman al firmei Sony, sînt costisitoare şi incomod de privit, la o diagonală de 4-5 cm. Ecranul virtual color ar fi o soluţie eficace.

După Allen Becker, preşedintele firmei RT şi inventatorul lui Private Eye, îmbunătăţirea rezoluţiei este iminentă, dar ecranul virtual în culori reale (full-colour) este încă de domeniul viitorului. Modelul ce va apare la finele lui 1991 va oferi o rezoluţie de 640x480 pixeli utilizînd tehnica de încapsulare existentă. Cel în culori va fi realizat atunci cînd vor fi disponibile comercial reţelele de LED-uri verzi şi albastre, la fel ca cele roşii din prezent.

Pentru primul model s-au ales LED-uri roşii tocmai deoarece sînt uşor de obţinut în reţele liniare şi au eficienţa luminoasă maximă. Oglinda basculantă va putea suprapune vizual pixelii celor trei culori fundamentale (RGB) ai reţelelor individuale de LED-uri pentru a produce culori multiple. Problema este că reţelele de LED-uri verzi şi albastre mai au cîţiva ani pînă să fie disponibile comercial, în variante cu eficienţă mare, fiabile şi ieftine.

(Cristian Malide, ICCE)

Comunicarea om-maşină

Nici în Evul High-Technology nu sînt realizate încă toate dorinţele oamenilor. O astfel de dorinţă încă nerealizată este şi maşina ca partener de comunicare. În lumea science-fiction-ului relaţia triunghiulară om-maşină- comunicare a fost rezolvată de mult. În realitate, oamenii de ştiinţă se ocupă de decenii cu această problemă, succesele fiind schimbătoare. Cine doreşte să afle care este starea de fapt, va putea arunca, prin intermediul acestui articol, o privire atît asupra fundamentelor cît şi asupra stadiului cercetărilor.

În anul 1952, IBM, a prezentat prima maşină care era capabilă să recunoască limbajul uman. Calculatorul uriaş, prezentat atunci, era în stare să recunoască numerele de la 1 la 10, citite cu voce tare, cu o precizie care atingea 97%. Această ştire de senzaţie a stîrnit o euforie enormă în rîndul cercetătorilor modului de comunicare om-maşină. Se credea că, în puţini ani, tastatura va putea fi înlocuită cu un microfon şi că operarea calculatoarelor se va putea realiza integral prin vorbire.

Acestei euforii i-a urmat curînd deziluzia, atunci cînd s-a constatat că algoritmul pentru recunoaşterea unor propoziţii complexe este cu mult mai complicat decît cel pentru recunoaşterea a zece numere şi necesită un efort mult mai mare.

Cercetările în domeniul recunoaşterii vorbirii au fost date, mai mult sau mai puţin, uitării. La sfîrşitul anilor '60, un studiu de la IBM a determinat guvernul SUA să susţină un proiect de cercetare, ceea ce a dat un nou avînt cercetărilor. Astăzi, o mulţime de institute şi universităţi studiază acest domeniu.

Cu toate că puterea de calcul a sistemelor actuale a atins cote gigantice, comparativ cu cea a anilor de început - şi s-ar putea crede că problema comunicării nu mai are mult pînă la a fi rezolvată, experţii apreciază posibilităţile concrete de realizare mai realist. Ei sînt de aceeaşi părere atunci cînd estimează ceea ce este realizabil într-un viitor previzibil şi ceea ce nu va exista niciodată: de exemplu un calculator care să poată înlocui în întregime interlocutorul uman.

Fundamentele recunoaşterii vorbirii

În prezent, comunicarea om-maşină se bazează pe aparatele de intrare: tastatură şi mouse, şi pe mijloacele vizuale de ieşire: monitor şi imprimantă. Ultimele pot fi înlocuite, între timp, fără probleme cu o ieşire bazată pe vorbire. Cu ajutorul unei plăci de extensie speciale şi al unui software specific, PC-ul poate vorbi pe orice ton dorit, de exemplu cu o voce joasă, bărbătească, sau cu una înaltă, feminină. Dacă primele PC-uri vorbeau încă monoton electronic, sistemele de ieşire moderne surprind prin accentul perfect şi intonaţia care respectă semnele de punctuaţie. Totuşi este mult mai simplu să se redea vorbirea, decît să fie recunoscută şi înţeleasă electronic.

În aceasta constă primul pas - şi astfel se iveşte şi prima problemă - pentru a putea recunoaşte electronic vorbirea; aceasta trebuie memorată (înregistrată) sub formă de text, respectiv trebuie transformată în informaţie digitală comprehensibilă pentru calculator. În timp ce un om nu are, cel mai adesea, probleme în a înţelege diferiţi vorbitori, calculatorului nu-i este la fel de uşor. Principial, limba vorbită nu este altceva decît o succesiune continuă de unde sonore de frecvenţe diferite. Astfel un cuvînt rostit poate fi reprezentat grafic înregistrînd variaţia frecvenţei undelor sonore în timp.

Probleme în cazul diferiţilor vorbitori

Comparînd fonogramele rezultate în urma înregistrării aceluiaşi cuvînt rostit de mai mulţi vorbitori, se observă că există diferenţe enorme. Acest lucru se datorează faptului că, de exemplu, femeile vorbesc folosind frecvenţe mai înalte decît bărbaţii. Un alt factor care determină diferenţele dintre fonograme este viteza de exprimare a vorbitorului. Datorită fiziologiei aparatului de vorbire uman, chiar şi în cazul în care acelaşi vorbitor rosteşte de mai multe ori acelaşi cuvînt, vor exista diferenţe între fonograme.

În plus trebuie luate în considerare şi zgomotul din mediul în care se vorbeşte, care şi el ajunge în microfon. De regulă, o rată de recunoaştere excelentă, de peste 95%, nu poate fi obţinută decît în condiţii de laborator, într-o încăpere izolată fonic.

Omul recunoaşte o propoziţie chiar şi atunci cînd unele cuvinte sînt articulate neinteligibil. Datorită cunoştinţelor generale pe care le posedă, ascultătorul poate înţelege, fără probleme, sensul propoziţiei. Dacă nu s-ar întîmpla aşa, atunci doi conaţionali vorbind dialecte diferite, nu s-ar putea înţelege.

Dacă va exista vreodată o maşină care să poată recunoaşte, la perfecţiune, vorbirea, atunci ea va trebui să dispună de o bază de date cu cunoştinţe generale. Ea este necesară, printre altele, şi pentru a permite înţelegerea "omofonelor". Omofonele sînt cuvinte care se pronunţă la fel, dar care au semnificaţii diferite, cum ar fi, de exemplu, "mai". Un partener de conversaţie uman deduce semnificaţia din context.

Două semnificaţii şi jumătate per cuvînt

În exemplele care urmează, datele statistice se referă la limba germană. Nu dispunem de date statistice referitoare la limba română. Dacă, întîmplător, acest articol va fi citit şi de un specialist care să ne poată pune la dispoziţie date statistice referitoare la limba română, ne vom bucura să revenim aducînd completările de rigoare.

Un cuvînt are, în medie, mai multe semnificaţii. Astfel în limba germană, "Dichtung", de exemplu, poate să însemne şi "poezie", dar şi "garnitură de etanşare" (a unui robinet). În limba română "mai" poate să fie adverb (mai ales), poate desemna o lună a anului, ficatul sau bucata de lemn cu care se bat rufele cînd se spală.

În cele din urmă, trebuie să se facă distincţie între vorbirea cursivă, numită şi vorbire legată, şi vorbirea cu pauze semnificative între cuvintele individuale, aşa-numita "vorbire cuvînt cu cuvînt". Recunoaşterea "vorbirii legate" poate fi realizată mult mai greu decît recunoaşterea "vorbirii cuvînt cu cuvînt". În cazul vorbirii legate este greu pentru sistem să recunoască cuvintele compuse. Sistemele de recunoaştere a vorbirii "cuvînt cu cuvînt" cer vorbitorului respectarea unor pauze clare între cuvinte. Vorbitorul este constrîns astfel să-şi adapteze, în mod nenatural, modul său de rostire necesităţilor maşinii.

Dialogul vorbit om-maşină este practic utilizabil numai dacă dialogul poate avea loc în limbajul uzual. Pe cît de

THE choice in software

AUTODESK
Microsoft
NOVELL
Lotus
Fox
BORLAND
Aldus
Corel
Symantec

雅仕 A&C INTERNATIONAL S.A.

Tel 40-0-53.53.15.

complexe sînt problemele de recunoaştere a vorbirii, pe atît de complexe sînt şi soluţiile utilizate în prezent.

În acest loc trebuie să mai atragem atenţia că între procesele de "recunoaştere a vorbirii" şi de "înţelgere a vorbirii" există diferenţe fundamentale, chiar dacă cele două domenii sînt strîns legate între ele. "A recunoaşte" înseamnă a converti în scris, sub formă de text corect d.p.d.v. sintactic, limba vorbită, deci sunetele ei individuale captate prin intermediul microfonului şi transmise calculatorului. "A înţelege" înseamnă a decela sensul şi conţinutul textului recunoscut, pentru a reacţiona în mod corespunzător. În cele ce urmează ne vom referi mai întîi la recunoaştere şi apoi la înţelegere.

Există trei tipuri de sisteme de recunoaştere a vorbirii: "sisteme dependente de vorbitor", "sisteme independente de vorbitor" şi "sisteme adaptive la vorbitor".

De la unda sonoră la modelul digital

Un prototip reprezentativ pentru stadiul actual al recunoaşterii vorbirii este sistemul de recunoaştere a vorbirii "Tangora", al lui IBM (denumit după Albert Tangora, campion mondial la scris cu maşina de scris). Tangora a fost prezentat pentru prima dată în SUA, în 1986. O echipă de cercetători IBM lucrează din 1988 în Heidelberg la o versiune germană, alte echipe lucrînd la versiuni pentru limbile italiană şi franceză. Tangora este un sistem de recunoaştere a vorbirii, dependent de vorbitor, destinat recunoaşterii propoziţiilor în cazul cuvintelor rostite izolat. Sistemul este funcţional pe calculatoarele PS/2 şi RS/6000 (sistem RISC cu sistem de operare UNIX - IBM), cu plăci de extensie speciale, procesoare de semnal şi memorii rapide. Explicaţiile care urmează se bazează pe principiul de funcţionare al sistemului Tangora. Principiul de funcţionare al hardware-ului de conversie a unui semnal analog (vorbirea) într-o formă inteligibilă pentru calculator este simplu: undele sonore corespunzînd fonogramei vorbitorului ajung prin intermediul unui microfon într-un convertor analog/digital. Vorbirea este recepţionată de acesta ca o succesiune continuă de frecvenţe diferite care pot fi reprezentate ca semnale digitale.

Eşantionare la 10 ms

Semnalul sonor este divizat, în acest convertor, în intervale de timp de 10 ms. Bucăţile astfel obţinute conţin un amestec de frecvenţe, care pot fi reprezentate în urma unei transformări Fourier sub formă de coeficienţi Fourier. Din acest spectru Fourier este prevalat un vector de caracteristici alcătuit din 20 de elemente, care este comparat cu 200 de vectori prototip, specifici vorbitorului. Simbolul care caracterizează prototipul cel mai apropiat este ataşat semnalului sonor. Astfel semnalul acustic este transformat într-o succesiune de simboluri, pe baza cărora poate demara procesul de recunoaştere.

Recunoaşterea limbii

Pentru ca un calculator să înţeleagă o limbă, trebuie să dispună de o bază de date în care să fie definite sunetele corespunzătoare. Memorarea fonogramelor tuturor cuvintelor individuale nu are sens atunci cînd se doreşte elaborarea unui sistem de recunoaştere a vorbirii pentru un vocabular vast (cca. 10.000 pînă la 20.000 de cuvinte), sistem care să fie independent de vorbitor şi adaptiv. Fiecare vorbitor ar trebui să rostească în faţa sistemului, de mai multe ori, fiecare cuvînt al vocabularului - operaţie care ar necesita foarte mult timp.

Mai avantajoasă este memorarea fonogramelor "fonemelor" decît a fonogramelor cuvintelor întregi. Fonemele sînt cele mai mici unităţi sonore ale limbii. Există cca. 70 de foneme, în limba germană fiind prezente cca. 60 dintre ele.

Pentru transcripţia limbii germane pentru sistemul de recunoaştere a vorbirii Tangora sînt utilizate 60 de foneme: trei foneme specifice sistemului: (Null-Phon, Linişte, Delimitator de propoziţie), 25 de vocale, 30 de consoane, două "offglides" (aşa sînt denumite sunetele diftongilor "au", "eu", "ei").

În cazul soluţiei cu memorarea setului de foneme apare problema coarticulării: sunetele individuale sînt puternic influenţate de sunetele învecinate, ceea ce face ca recunoaşterea lor să fie dificilă şi nesigură.

Din acest motiv este de preferat memorarea unor unităţi sonore mai mari. În cazul unităţilor mai mari - în practică fragmente de silabe - pot fi cuprinse şi efectele datorate coarticulării şi se poate ţine seama de ele. În baza de date, cuvintele care urmează a fi recunoscute sînt definite ca succesiuni de astfel de unităţi sonore. Definiţiile sînt date cu ajutorul aşa-numitelor "modele Hidden-Markov". Descris simplificat, un model Hidden-Markov constă dintr-un număr de stări, între care sînt posibile tranziţii. Pentru fiecare stare sînt definite tranziţiile posibile. În plus, pentru fiecare stare există o anumită probabilitate de ieşire, care modelează probabilistic succesiunea de unităţi sonore (foneme). Dacă vorbitorul rosteşte un cuvînt, sistemul recunoaşte, mai întîi, acest cuvînt ca pe o succesiune probabilă de unităţi sonore (corespunzînd stărilor în modelul Hidden-Markov). În timpul analizei, este căutat, în baza de date, cuvîntul definit ca model Hidden-Markov, care se potriveşte cel mai bine cu înregistrarea.

În baza de date vor fi găsite mai multe sute de cuvinte care vor candida pentru identificare, dintre acestea fiind ales acela care corespunde cuvîntului rostit cu cea mai mare probabilitate.

Fast-Match

Prima metodă de căutare, denumită "Fast-Match", constă din mai mulţi paşi de verificare, în care numărul de cuvinte este redus în urma unor examinări din ce în ce mai exacte. Atunci cînd nu mai rămîn decît cca. 100 de cuvinte posibile, urmează următoarea etapă de analiză. În cadrul acestei etape se accede o culegere de texte, în sistemul Tangora, gigantică. Această culegere de texte a fost realizată, de colectivul de cercetători care se ocupă de sistemul Tangora, prin citirea automată (scanner + software OCR) din reviste economice. Momentan, culegerea de texte Tangora cuprinde cca. 50 de milioane de cuvinte, şi creşte în fiecare lună cu cca. un milion de cuvinte.

Această culegere de texte a fost realizată în scopul determinării statistice a frecvenţei de apariţie a unor succesiuni de două sau de trei cuvinte. În acest fel numărul de cuvinte care rămîn în competiţie este redus din nou.

Cuvintele rămase sînt supuse unei analize denumită Detailed-Match. Analiza Detailed-Match se aseamănă cu analiza Fast-Match, dar ea nu mai operează cu fonemele modelelor Hidden-Markov ci cu unităţi sonore artificiale, aşa-numite "foneme". Cu ajutorul acestora se poate efectua o verificare acustică mai exactă. După parcurgerea analizei Detailed-Match, fraza este, în fine, recunoscută. Versiunea germană a sistemului Tangora recunoaşte, în prezent, 13.000 de cuvinte cu o probabilitate de peste 95%. În alte limbi sînt înţelese 20.000 de cuvinte cu o probabilitate de peste 95%.

De la a recunoaşte, la a înţelege

Recunoaşterea corectă a vorbirii este doar unul dintre aspectele importante pentru comunicarea în mod natural dintre om şi maşină. Mult mai important, şi în acelaşi timp mai greu, este să aduci calculatorul să înţeleagă conţinutul, sensul, a ceea ce a recunoscut.

Pentru o maşină care trebuie să scrie după dictare poate să fie suficient să recunoască şi să transcrie sub formă de text ceea ce i se spune. O comandă nu poate fi executată însă decît dacă i se înţelege semnificaţia. Imaginaţi-vă un robot care să se poată deplasa asemănător omului. El dispune de ochi (cameră de luat vederi) şi urechi (microfoane). Omul care îl comandă ţine legătura cu el prin radio, dispune de un monitor pe care se afişează ceea ce "vede" robotul şi îi comunică în limbaj natural ceea ce are de făcut. În cazul comenzii "Du-te trei metri spre nord", robotul nu va avea probleme. Şi nici în cazul în care i se va spune "Du-te la masă", cu condiţia ca robotul să se afle într-o încăpere în care să existe o masă, să dispună de o cameră de luat vederi, de un program de recunoaştere a formelor şi în baza lui de date să fie definit modul în care arată o masă.

Să presupunem că robotul este folosit ca reporter şi că este dotat cu un aparat de fotografiat. El se găseşte într-o încăpere în care se află mai multe persoane şi primeşte comanda. "Fotografiază bărbatul cu aparatul de fotografiat". Soft-ul cu care este dotat robotul analizează propoziţia şi încearcă să-i determine sensul. Comanda de bază, "fotografiază" este uşor de înţeles; tot la fel şi obiectul la care se referă această comandă: "bărbatul". Sensul ultimei părţi a propoziţiei "cu aparatul de fotografiat" nu poate fi stabilit exact. Dacă în încăpere se găseşte un singur bărbat, atunci este clar că robotul trebuie să fotografieze acest bărbat cu aparatul de fotografiat.

Ce se întîmplă însă atunci cînd în încăpere se află mai mulţi bărbaţi şi unul dintre ei are atîrnat la gît un aparat de fotografiat? Fără îndoială că, în acest caz, robotul trebuie să-l fotografieze pe acel bărbat. Pentru ca un robot să poată înţelege sensul unei instrucţiuni (comenzi), nu este deci suficient să fie în stare să descompună o propoziţie în părţile sale componente. El mai are nevoie, în plus, de o bază de cunoştinţe în care, pentru exemplul de mai sus, să fie definit ce este un bărbat şi ce operaţie este desemnată ca fotografiere. Instrucţiunea "fotografiază" ar putea fi descrisă printr-o succesiune de operaţii de genul următor: "ia un aparat de fotografiat, îndreaptă obiectivul în direcţia obiectului de fotografiat, ţine-l nemişcat şi declanşează". Dar nici această bază de cunoştinţe nu este încă suficientă pentru exemplul anterior. Robotul trebuie să mai ştie că instrucţiunea "fotografiază obiect" este suficientă şi că este de prisos să se facă specificarea "cu ce" doar întotdeauna se fotografiază cu un aparat de fotografiat şi nu cu altceva. Dacă specificarea suplimentară este prezentă, ea nu poate servi decît la a descrie mai exact obiectul vizat (bărbatul care are un aparat de fotografiat). Pentru o înţelegere perfectă robotul ar trebui să dispună de o bază de date care să corespundă cunoştinţelor umane - dar, cu capacităţile actuale de memorare, aşa ceva nu este posibil. Şi chiar dacă ar exista capacitate de memorare suficientă, cine ar putea defini baza de cunoştinţe? Robotului anterior descris i-ar fi cu siguranţă greu să execute comanda care i s-a dat. Vinovat de această situaţie este cel care a dat comanda. El ar fi trebuit să-şi exprime dorinţa mai puţin ambiguu, de exemplu: "fotografiază bărbatul care are atîrnat la gît un aparat de fotografiat".

Confuzii fără limite

Chiar şi în cazul celor mai mari baze de cunoştinţe, cel care dă comanda este forţat de maşină să-şi formuleze comenzile cu o precizie pe care omul n-o foloseşte decît în cele mai rare cazuri. El poate să-şi formuleze verbal comanda, dar trebuie să ţină cont de limitările maşinii. În aceste condiţii, comunicarea om-maşină, în limbaj natural, nu are sens. O soluţie acceptabilă, în acest domeniu, va exista doar atunci cînd maşina va înţelege limbajul curent. Cît de repede se va realiza acest lucru? Asupra acestui punct de vedere, cercetătorii sînt de aceeaşi părere - vor trece decenii. Şi probabil că nu va exista niciodată o soluţie mulţumitoare. Realizabile sînt însă sistemele de comunicare în limbaj natural pentru anumite domenii de aplicare, pentru care să se memoreze cunoştiinţele lingvistice şi conceptionale. Un exemplu în acest sens este sistemul de comandă în limbaj natural "Dos-Man" oferit de firma Transmodul Software din Saarbrucken, începînd din 1989 şi care este perfecţionat în continuare. DOS-Man este un DOS-Shell, cu ajutorul căruia puteţi opera la calculator, dînd comenzi sistemului de operare în limbaj natural. În locul comenzii "dir *.txt", în cazul lui Dos-Man trebuie dată comanda: "afişează toate fişierele cu extensia txt" (în germană). În plus, pot fi date şi comenzi mai complexe, cum ar fi, de pildă, "afişează toate fişerele care au fost create în această săptămînă", sau "caută programul Word".

În continuare vom descrie modul de funcţionare al lui Dos-Man. Analiza unei propoziţii se face în cinci paşi: "scanare", "analiză morfologică", "analiză sintactică", "analiză semantică" şi "analiză pragmatică".

Scanare

Propoziţiile rostite sînt atribuite mai întîi unor variabile text. Scanner-ul desface o propoziţie - care în acest moment pentru Dos-Man nu este decît o succesiune de caractere - în cuvinte individuale. Depistarea cuvintelor din propoziţie se face prin căutarea spaţiilor dintre caractere. În plus, sistemul examinează şi caracterele care nu sînt litere. Pentru descrierile ulterioare să considerăm comanda: "afişează telexurile, mai vechi de o lună, în ordinea dimensiunii".

Analiza morfologică

În cadrul analizei morfologice, cuvintele obţinute în urma scanării sînt aduse în forma lor normală, de exemplu verbele sînt puse la infinitiv, iar substantivele la nominativ singular. Pentru aceasta sistemul accede dicţionarul şi, pe baza terminaţiilor, stabileşte clasa cuvîntului şi obţine şi alte informaţii lingvistice. Astfel "afişează" devine "a afişa" şi i se adaugă informaţiile suplimentare "verb, imperativ", iar "telexurile" devine "telex" cu informaţiile suplimentare "substantiv, plural".

Deoarece în limba germană fiecare cuvînt poate avea în medie 2,5 semnificaţii şi poate fi încadrat în tot atîtea clase, analiza este mult mai complicată decît pare la prima vedere. În urma analizei morfoloice sistemul nu poate stabili încă, dacă, de exemplu, cuvîntul "unu" este un articol sau un număr.

Analiza sintactică

În cadrul analizei sintactice sînt determinate părţile de vorbire ale propoziţiei, cu ajutorul unei gramatici care conţine o mulţime de reguli ce definesc structura limbii. Pentru exemplul anterior, în urma analizei sintactice va rezulta: "predicat: afişează, subiect: telexurile, atribut participial: în ordinea dimensiunii, atribut relativ: mai vechi de o lună".

Analiza sintactică returnează propoziţiile incorecte sintactic, respectiv care

nu respectă regulile gramaticale, cum ar fi, de exemplu, "mai vechi de afişează şi telexurile o lună".

Analiza semantică

Scopul analizei semantice este determinarea sensului unei propoziţii. Pentru aceasta sînt necesare cunoştinţe lingvistice. Pe de altă parte sînt necesare cunoştinţe conceptuale. De exemplu, trebuie ştiut că fişierele care conţin telexuri au extensia "tlx" sau că instrucţiunea "mai vechi de o lună" înseamnă că diferenţa dintre data de creare a fişierelor respective şi data curentă trebuie să fie mai mare de o lună. Baza de cunoştinţe conceptuale trebuie să permită, în plus, recunoaşterea unor comenzi lipsite de sens; cum ar fi "şterge calculatorul". În baza de cunoştinţe sînt înregistrate ca obiecte care pot fi şterse fişierele şi directoarele, nu şi calculatorul. Pentru exemplul anterior, în urma analizei semantice, se va obţine următorul rezultat:

- afişează, cum: în ordinea dimensiunii
- ce: fişierele cu extensia ".tlx"
- care - data de creare < data curentă - o lună

Analiza pragmatică

În cadrul analizei pragmatice, structurile semantice sînt puse în corespondenţă cu comenzile sistemului de operare:

1. determină directorul curent
2. determină fişierele cu extensia ".tlx" din acest director
3. determină care dintre aceste fişiere au data de creare conform cerinţei
4. sortează aceste fişere după dimensiune
5. afişează lista de fişiere sortată.

În acest mod comenzile devin inteligibile şi pentru calculator şi pot fi executate pas cu pas.

Fundamentele şi problematica metodelor de eşantionare

Un sistem care poate analiza instrucţiuni verbale este denumit analizor (parser). Procedura de analiză se compune din trei paşi: analiza sintactică, analiza semantică şi analiza pragmatică. În timpul analizei există două probleme dominante. Una dintre ele constă în descrierea formulării de propoziţii cu reguli precise - gramatica cu care poate fi creată orice propoziţie imaginabilă într-o anumită limbă. Cealaltă problemă constă în analiza propriu-zisă. Propoziţiile simple, constînd din verb - adjectiv - substantiv, pot fi prelucrate uşor. Lucrurile se complică în cazul frazelor complicate, cum ar fi: "caută în dulapul mic, din camera de zi, toate scrisorile care au fost puse acolo în urmă cu două săptămîni". În acest caz cuvîntul "acolo", care se poate referi atît la "dulap" cît şi la "cameră", provoacă dificultăţi. Un analizor (parser) nu poate prelucra, de la început, o frază în întregime. El este obligat să examineze părţile de propoziţie în parte, una după alta. Dacă s-a prelucrat doar o parte a frazei, atunci analizorul poate înţelege acea parte, dar îi este greu să-şi dea seama ce rol joacă această bucată pe asamblu şi cît de importantă este pentru înţelesul final.

Întrucît o propoziţie aflată la sfîrşitul frazei poate avea o influenţă decisivă asupra semnificaţiei ei, soluţia de a analiza o frază cuvînt cu cuvînt nu este neapărat cea mai bună. În prezent există analizoare care lucrează după următoarele principii: "descindere recursivă", "acceptor final" şi "Noise-Disposal".

Descindere recursivă

În cazul acestei metode, analizorul examinează mai întîi elementele din care este alcătuită fraza şi divide aceste elemente în continuare, pînă cînd se ajunge la elemente care nu mai pot fi divizate în continuare. Mai simplu spus, un astfel de analizor trebuie să ştie că din punct de vedere gramatical, o propoziţie constă dintr-o parte nominală şi dintr-una verbală. În afară de aceasta, el trebuie să dispună de o bază de date în care să fie stocate semnificaţiile gramaticale ale cuvintelor, de exemplu "un = articol", "hoţ = substantiv". Alte reguli gramaticale vor stabili, de exemplu, că un articol poate însoţi numai un substantiv sau un adjectiv, pentru ca propoziţia să aibă sens.

Acest mecanism funcţionează totuşi numai în cazul propoziţiilor simple. Regulile gramaticale ale unei limbi sînt în mod firesc prea complexe pentru a putea fi descrise inteligibil pentru un analizor. În plus, nu este suficient să i se spună analizorului cărei părţi de vorbire îi aparţine fiecare cuvînt.

Ca exemplu se dă de obicei propoziţia "Brînză mănîncă şoarecele". Această propoziţie este înţeleasă foarte uşor de către om. Unui om nu i-ar trece niciodată prin cap că şoarecele ar putea fi cel mîncat. Un analizor ar putea înţelege această propoziţie numai cu condiţia ca în baza sa de cunoştinţe să existe nişte definiţii de genul următor:

- brînză (rigid, aliment)
- şoarece (fiinţă)
- a mînca (a consuma alimente)

Ce s-ar întîmpla însă dacă analizorul ar trebui să înţeleagă că un burghiu muşcă cîte puţin din peretele de beton?

Acceptor final

Un astfel de analizor examinează mai întîi primul cuvînt din frază şi cu ajutorul lexiconului de cuvinte îi determină semnificaţia gramaticală (de exemplu "un=articol"). Principiul acceptorului final constă în faptul că o stare iniţială se transformă, prin intermediul unei anumite reprezentări, într-o stare finală posibilă. Stările posibile sînt definite de părţile de vorbire ale cuvintelor din propoziţie.

În exemplul anterior, prima stare este un articol. Elementele posibile care îl pot succeda sînt definite cu ajutorul unui graf. În acest graf se va stabili, de exemplu, că un articol nu poate fi urmat decît de un adjectiv sau de un substantiv. Analizorul parcurge deci fraza cu ajutorul acestui graf. Problema care apare în cazul acestui tip de analizor este că analiza eşuează imediat, dacă o propoziţie nu este corectă gramatical sau dacă gramatica a fost insuficient descrisă în graf. Astfel apare aceeaşi problemă ca şi în cazul descinderii recursive, şi anume necesitatea definirii exacte a unei gramatici.

Noise-Disposal

Analizoarele Noise-Disposal nu ţin cont de gramatică, ci doar de conţinutul unei propoziţii. Toate cuvintele din propoziţie pe care nu le înţeleg sînt privite ca "zgomot" (noise) şi sînt "îndepărtate" (to dispose) pur şi simplu din propoziţie. Din cuvintele rămase se încearcă obţinerea unui sens. Este clar astfel că principiul de eliminare a zgomotelor nu se pretează pentru o comunicare exactă, în limbaj natural, între om şi maşină. Principiul se poate dovedi totuşi bun pentru comunicări simple, în limbaj natural, desigur, cu condiţia ca analizorul să dispună de un tezaur de cuvinte suficient de mare pentru domeniul de aplicare vizat.

(R.M.)

Bibliografie:

- *Michael Nickles: Mensch-Maschine-Komunikation, DOS, iulie 1991, DMV- Verlag, ISSN 0933-1557*
- *Reiner J. König: Der Computer hört aufs Wort, Computer persönlich, 24/91, Markt & Technik Verlag, ISSN 0722-0987*

We draw on your imagination.
A Bold New Shape.
A Fast New Pace.
smooth. Our
thanks to
and
Take a bold first step to reshape plotting at your firm. Find out more today: Call 800-932-1212. In Canada, call 416-635-9010. We're expecting to hear from you.
A & C
INTERNATIONAL S.A.
雅仕
AUTHORIZED DEALER
Constantin Tanase Str.
No. 15, Sect. 2,
Bucharest, 73299
Romania
Tel: 40-0-53.53.15
Service - "Hot Line": 40-0-12.77.73
Fax: 40-0-12.77.74
Telex: 10674 MAXIM R
CalComp
A Lockheed Company

Guru 3.0

Construirea de sisteme multimedia

Cea mai nouă versiune 3.0 a pachetului de dezvoltare sisteme expert Guru oferă proiectanţilor posibilitatea de a-şi defini funcţii proprii şi de a memora şi prelucra fişiere cu imagini, grafică sau secvenţe sonore cu ajutorul aşa-numitelor BLOB (Binary-large-objects).

Guru a fost creat în octombrie 1985 şi este unul din primele instrumente de dezvoltare sisteme expert. Produsul a fost, şi a rămas, cu un pas înaintea timpului său.

Sistemul cuprinde modulele: bază de date, calcul tabelar, grafică de afaceri şi editor de texte. Prin intermediul unei interfeţe de comunicaţie datele pot fi extrase de pe un calculator mai mare sau, de ex., pot fi interogate cursurile de la bursă. Toate modulele sînt complet integrate, astfel încît din modulul de bază de date se poate lansa o comandă pentru producerea unui grafic, sau din cadrul programului de calcul tabelar se poate lansa o interogare a sistemului expert.

Guru dispune de generatoare de măşti şi de rapoarte flexibile; limbajul de interogare al bazei de date este compatibil SQL.

Proiectarea de aplicaţii se realizează prin intermediul unui limbaj de programare puternic, din categoria 4GL. O interfaţă aşa-numită în limbaj natural permite comunicarea cu Guru în limbajul uzual, desigur doar în engleză.

Maşina de inferenţe

Guru memorează cunoştinţele expert în forma clasicelor reguli "Dacă - Atunci".

În dezvoltarea de sisteme expert shell-urile bazate pe reguli pierd tot mai mult teren. Din acest motiv li se reproşează adesea proiectanţilor mediului Guru, faptul că mizează atît de mult pe sistemele bazate pe reguli. Totuşi Guru dă posibilitatea formulării unor reguli foarte complexe, care să permită o bună prelucrare a cunoştinţelor experţilor - după ce acestea au fost strict formalizate.

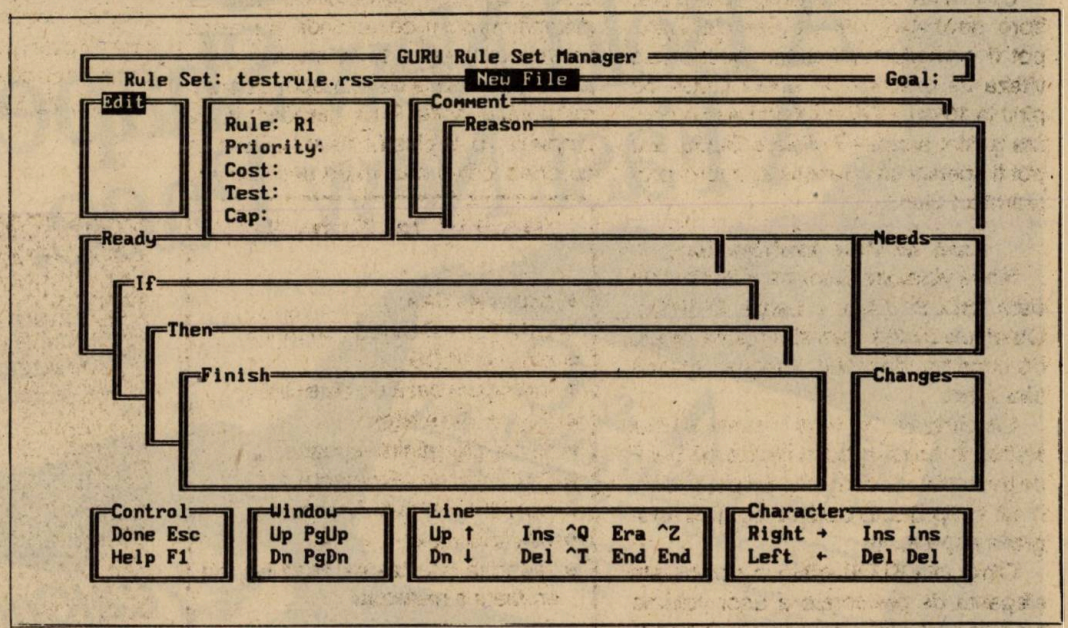
Maşina de inferenţe a shell-ului Guru prelucrează regulile înlănţuit înainte şi înapoi. Modul de căutare poate fi influenţat; cunoştinţelor vagi li se pot ataşa factori de probabilitate şi ele pot fi prelucrate cu ajutorul logicii Fuzzy.

O nouă situare pe piaţă

Guru este o extensie a lui "Knowledge Man" care este desemnat ca Decision-Support-Systems. Multă vreme nici în cazul lui Knowledge Man nu a fost clar dacă era vorba despre un sistem destinat proiectanţilor sau utilizatorilor finali. Din această cauză Knowledge Man apărea uneori în teste comparative cu baze de date, alteori în teste comparative cu pachete integrate. Întotdeuna a fost lăudată puterea produsului, dar a fost criticat modul de operare complicat.

Asemănătoare a fost situaţia şi în cazul lui Guru. Odată cu noua versiune 3.0, proiectantul - Micro Database Systems (MDBS) Inc. - defineşte explicit Guru ca instrument de dezvoltare. Acest lucru devine clar şi prin prezentarea limbajului de programare de nivel 4GL, care acum se numeşte KGL (KnowledgeMan-Guru-Language). Datorită deplasării centrului de greutate spre programare, manualele au trebuit rescrise şi restructurate. Documentaţia este prezentată sub forma unui manual de referinţă pentru KGL şi a unei prezentări de ansamblu a modulelor. Preţioasă este o nouă cărţulie care se ocupă în mod special cu tehnica dezvoltării sistemelor expert.

Repoziţionarea a implicat şi pierderea suprafeţei utilizator clasice. În locul ei a fost creată o nouă suprafaţă SAA



Suprafaţa utilizator Guru: masca de intrare pentru formularea de reguli pentru sistemul expert

(scrisă în KGL şi disponibilă în cod sursă, pentru a servi de exemplu ca program bine scris în KGL).

Printre avantajele noii suprafeţe se numără posibilitatea de a vedea oricînd modul de ocupare al memoriei, sau de a interoga o bază de date prin metoda Query-by-exemple. Dezavantajul în constituie faptul că parcursul prin punctele de meniu nu mai este transformat, ca la versiunile anterioare, în comandă şi nu mai este prezentat ca linie de comandă. Această abordare uşura învăţarea limbajului de comandă.

În timpul testelor, modul de lucru cu meniuri a provocat de două ori pierderi de date. La ieşirea din modulul calcul tabelar nu s-a pus întrebarea dacă datele trebuie salvate şi în ce fişier; memorarea lor n-a decurs corect şi un nou apel al tabelei tocmai prelucrate prin meniu n-a mai fost posibil. În modul comandă, în schimb Guru a reacţionat corect întotdeauna.

Funcţii definite de utilizator

Cu ajutorul comenzii "perform procedure" sînt apelate şi prelucrate programe Guru stocate în fişiere. Pînă acum transferul valorilor (parametrilor) spre programe şi între programe se făcea cu dificultate. Un tip de funcţie aşa-denumit "User-defined-functions" (UDF) permite aşa- numitul "parameter-passing", aşa cum este uzual în alte limbaje de programare.

UDF-urile sînt subprograme care, spre deosebire de programele Guru, pot fi compilate. În acest mod creşte viteza de execuţie, în unele cazuri de pînă la 10 ori. UDF-urile primesc un nume şi sînt stocate în fişiere BLOB. Ele pot fi apelate ca o funcţie în cadrul programelor Guru.

Bază de date multimedia

Noua versiune Guru permite tipul de date "BOLB" (Binary Large Objects). Obiectele binare mari sînt fişiere binare de orice tip, deci text, imagine, grafică sau sunet.

Un cîmp BLOB este o parte a unui articol. În acest mod în datele de bază de personal, de exemplu, se poate defini un cîmp BLOB care să conţină fotografia angajatului.

Cîmpurile BLOB oferă o posibilitate elegantă de prelucrare a unor volume mari de date nestructurate. Cu Guru pot fi create, de exemplu, arhive de imagini sau arhive muzicale. Dacă se mai adaugă şi inteligenţa tehnologiei sistemului expert, Guru ar putea crea, de exemplu, o arhivă cu bucăţile muzicale transmise de un anumit post de radio pentru un anumit profil al ascultătorului şi/sau între anumite ore din zi.

Singura dificultate: este necesar un software corespunzător pentru reprezentarea graficii, textelor, etc., memorate într-un cîmp BLOB. De exemplu, cine prelucrează cu Guru fişiere cu grafică Hervard Graphics, trebuie să apeleze HG de sub Guru.

Harvard Graphics va afişa în acest caz imaginea dorită. Fişierele PIC pot fi afişate de Guru şi fără ajutorul altui program.

Prelucrarea datelor cu cîmpuri BLOB mai pune o problemă: volumele mari de date.

Găsirea unei bucăţi muzicale sau a unei imagini pe suportul de date se întîmplă în zecimi de secundă, numai că fişierele cu imagini sau sunete necomprimate au nevoie de mult spaţiu pentru memorare, astfel încît harddisk-ul se umple rapid. În plus un fişier BLOB este cu cîţiva octeţi mai mare decît fişierul original, deoarece sînt adăugate informaţii de identificare. CD-ROM este suportul de memorare ideal pentru bazele de date multimedia create de Guru.

Mai rapid, mai puţin spaţiu de memorie

Codul sursă al programului Guru a fost optimizat şi comprimat în permanenţă de la versiunea 1.1 încoace. Deşi se recomandă o memorie de 640 KByte, versiunea 3.0 - sub "Deskview" de exemplu - se mulţumeşte şi cu 360 KByte.

Un sistem de consultanţă creat cu versiunea 1.1 a fost rulat în versiunea 3.0 cam de două ori mai rapid. Guru este deplin funcţional, deci, chiar şi pe un simplu XT. Pe harddisk ocupă cca. 5 MByte.

Mediul de dezvoltare este disponibil pentru OS/2, VAX, Sun şi Unix.

Guru 3.0 costă 12.000 DM pentru un utilizator, 30.000 DM, versiunea LAN (număr nelimitat de utilizatori per Server). Există şi un sistem de dezvoltare pentru începători la preţul de 3.195 DM. Modulul Runtime costă 1.600 DM.

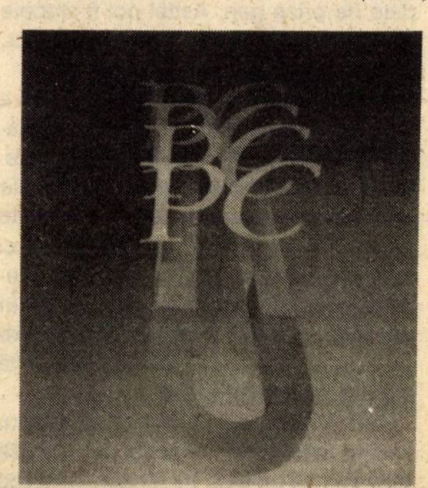
(R.M.)

Sisteme expert create cu Guru

- **Volkswagen AG, Wolfsburg**: Prognoză asupra disponibilităţii personalului
- **Pflanzenschutzamt Hannover**: Consultanţă privind utilizarea substanţelor chimice pentru protecţia plantelor
- **Istitution Quimico de Sarria, Barcelona**: Sistem expert privind prelucrarea petrolului
- **Nasa Ames Research Center**: Suport utilizator Cray pentru o mai bună coordonare a reţelei de supercalculatoare.
- **Army Secretary for Joint Actions**: Prelucrarea şi regăsirea documentelor.
- **Union Bank Finnland**: Sistem expert de consultanţă în probleme de finanţare.

Noutăţi la Guru 3.0

- suprafaţa SAA
- UDF (User Defined Functions)
- cîmpuri BLOB
- interogare bază de date QBE
- Report Templates
- limbaj de definire rapoarte
- SQL - Server - Frontend
- necesită mai puţină memorie
- este mai rapid
- permite analizarea modului de ocupare a memoriei



Blob, blob...

Un nou tip de cîmp pentru bazele de date va schimba scena din temelii: "Binary Large Objects", denumite şi BLOB, oferă utilizatorului de baze de date o mulţime de noi posibilităţi.

Lumea informaticii ar fi de neimaginat, astăzi, în absenţa bazelor de date relaţionale. Multe sisteme utilizează limbaje de programare proprii, înglobate în sistem cum sînt de pildă limbajele de nivel 4GL, sau limbaje de programare tradiţionale, cum ar fi C, Cobol şi Fortran. O caracteristică distinctivă a bazelor de date o constituie organizarea datelor sub formă de tabele: datele sînt prelucrate pornind de la o structurare în rînduri (articole) şi coloane (cîmpuri). De regulă se utilizează cîmpuri şi articole de lungime fixă.

Ca tipuri de date se utilizează cîmpurile de tip şir, tip numeric, respectiv logic ca şi tipurile derivate cum ar fi: dată, timp, cîmp întreg. Unele sisteme utilizează şi cîmpuri speciale, în care se pot memora texte sau grafice de lungime variabilă. Doar în puţine sisteme este prezent tipul de date, relativ nou, "Binary Large Objects", mai scurt denumit şi BLOB.

BLOB-urile reprezintă mai mult decît cîmpurile tip Memo normale

BLOB-urile au o lungime fixă. Spre deosebire de cîmpurile de tip Memo sau de cîmpurile conţinînd grafică, cîmpurile BLOB pot conţine nu doar texte ASCII sau doar grafice. Ele permit stocarea, în contextul unui articol, a unui volum de date de orice gen. Astfel pot fi stocate imagini grafice, sunet (vorbire, zgomot, muzică), şi desigur şi texte.

Întru-cît tipul de date care urmează a fi stocate, nu trebuie stabilit cu exactitate, BLOB-urile sînt universal utilizabile. Fizic, în mod normal, ele nu sînt stocate în aceeaşi structură de date cu celelalte cîmpuri ale articolelor bazei de date, ci în fişiere distincte. Schiţa pe care o prezentăm va lămuri această problemă. În articolul propriu-zis se stochează doar un pointer, care va indica poziţia exactă în cadrul fişierului BLOB.

Din motive de performanţă, pentru fiecare intrare într-un astfel de fişier se rezervă cîte un bloc de lungime fixă (de ex. 512 Byte). Atunci cînd se actualizează conţinutul unei anumite intrări BLOB şi dimensiunea creşte astfel încît blocul rezervat nu mai este suficient pentru stocare, fişierul BLOB trebuie reorganizat intern. Această reorganizare are loc în detrimentul vitezei de prelucrare.

Dacă dimensiunea standard a blocurilor utilizate poate fi liber stabilită, dimensiunea lor este determinată principial de necesarul de memorie mediu, sau maxim, pentru ceea ce trebuie memorat. Principial, totuşi, conţinutul unui BLOB nu este limitat în dimensiune şi poate atinge domenii de ordinul gigaocteţilor.

Multimedia şi BLOB

Progresul tehnic a făcut ca expresii ca "multimedia" sau "prelucrarea imaginilor" să fie pe buzele tuturor. În multe aplicaţii sînt integrate şi imagini, prin referinţe la numele corespunzătoare de fişiere. Un sistem care conţine puţine imagini sau grafice le poate trata în mod onorabil. Dacă biblioteca grafică creşte în volum, atunci creşte enorm şi efortul depus pentru prelucrarea, organizarea şi regăsirea lor.

Pentru prelucrarea şi regăsirea de imagini şi grafice cu ajutorul unei baze de date, tipul de date BLOB oferă premise excelente. Cu ajutorul lui nu este nici o problemă să se formuleze o interogare în limbaj SQL (o comandă SELECT, de ex.) asupra unei biblioteci grafice. Condiţiile impuse se pot referi, de exemplu, la texte stocate în alte cîmpuri ale articolului. Dar a rezuma posibilităţile de utilizare a cîmpurilor tip BLOB doar la aplicaţiile multimedia nu ar fi corect.

Guru, Informix şi KnowledgeMan

Printre aplicaţiile comerciale care integrează deja cîmpurile de tip BLOB se numără, printre altele, şi produsele elaborate de specialistul în baze de date MDBS: "Guru" şi "KnowledgeMan" ca şi "Informix - OnLine" al firmei Informix Software. La ambele produse MDBS pot fi integrate oricîte cîmpuri de tip BLOB în cadrul unei tabele. Limbajul de programare integrat "KGL" dispune de o mulţime de funcţii care permit memorarea, respectiv citirea, de date din cîmpurile BLOB. Aceste date pot fi orice tip de fişiere externe (texte sau binare, incluzînd fişiere executabile tip EXE sau COM), sau şi şiruri interne, matrici uni sau bidimesnionale; formate grafice interne, tipuri de date definite de utilizator. În plus variabilelor li se pot ataşa mai multe valori şi anumiţi coeficienţi de probabilitate (variabile fuzzy). În SGDB-urile tradiţionale aceste tipuri de date nu pot fi utilizate decît cu un efort considerabil. Cu ajutorul BLOB-urilor problema devine trivială. Ca exemplu ar putea servi următoarea comandă Guru/KnowledgeMan:

```
SELECT DISPLAY (PERSONAL.IMG)
FROM PERSONAL
WHERE SEX = 2 AND AGE<=30
```

Această comandă SQL va determina afişarea pe monitor a fotografiilor tuturor angajatelor înregistrate în tabela PERSONAL, care au vîrsta mai mică sau egală cu 30 de ani.

Există şi posibilitatea stocării unor secvenţe de program, anumite rutine, funcţii de bibliotecă şi a altor obiecte în cîmpurile BLOB, care să fie utilizate apoi în proiectarea de software sau la execuţia unei aplicaţii.

Un efect colateral suplimentar îl constituie extinderea mecanismelor de protecţie (drepturi de scriere/citire) specifice bazelor de date, şi asupra acestor tipuri de cîmpuri. În acest mod şi aceste date pot fi protejate împotriva unor accedări neautorizate.

Dacă SGBD-ul utilizat dispune de interfeţele şi de funcţiile necesare atunci nu mai există nici o predică în calea materializării ideilor utilizatorului.

Scena bazelor de date se modifică

Industria SGBD-urilor începe abia acum să comercializeze tehnologia BLOB. Ingres, în postura sa de lider, şi mulţi alţii, şi-au făcut publică intenţia de a integra cîmpurile de tip BLOB în arhitecturile proprii. BLOB-urile vor influenţa în mod decisiv modul de proiectare al pachetelor software bazate pe baze de date şi al sistemelor expert. Toţi proiectanţii de SGBD-uri relaţionale vor trebui, mai curînd sau mai tîrziu, să includă în produsele proprii tipul de date BLOB.

(continuare în pag. 24)

Baze de date orientate obiect

După limbajele de programare orientate obiect vin la modă bazele de date orientate obiect. Ce sînt aceste baze de date şi care sînt performanţele lor va fi exemplificat printr-un proiect care funcţionează în cadrul Societăţii pentru Matematică şi Prelucrarea datelor (GMO) din Germania, în momentul de faţă.

Sistemele de gestiune a bazelor de date (SGBD) joacă un rol foarte important în prelucrarea şi administrarea unor cantităţi mari de date. Importantă nu este doar eficienţa serviciilor caracteristice ale SGBD-urilor cum ar fi managementul tranzacţiilor, protecţia la acces, siguranţa datelor şi integritatea lor ci şi pretarea modelului de date ales la modelarea datelor. Să presupunem că avem o mulţime mare de date structurate în mod complex, de exemplu structuri de documente şi formulare, de layouturi sau/ şi structuri geometrice complexe pe care trebuie să le memorăm într-o bază de date relaţională. În acest caz structurile complexe vor fi modelate ca o mulţime de structuri plane sub forma unor relaţii. Deci unităţile legate între ele printr-o anumită logică vor fi descompuse în unităţi mai mici. Astfel diferitele programe utilizator vor trebui să compună în mod greoi, prin paşi succesivi, de interconectare, (operaţii joint), din fiecare relaţie plană obiectele complexe sau să implementeze operaţiile logice pe obiectele complexe ca şi operaţii pe relaţii plane. Acest lucru nu este critic doar din punct de vedere al eficienţei diferitelor programe utilizator, ci şi în ceea ce priveşte interpretările şi manipulările consistente asupra structurilor plane, care reprezintă de fapt obiecte complexe. SGBD-ul poate garanta doar consistenţa pe structurile plane cunoscute lui şi în mai mică măsură consistenţa obiectelor structurate complex.

Un SGBD orientat obiect, cum ar fi cel din proiectul VODAK de la Institutul pentru Sisteme Integrate Informaţionale şi Publicistice, oferă concepte şi mecanisme, care evită asemenea dezavantaje şi limitări şi care simplifică administrarea şi prelucrarea unei cantităţi mari de date structurate. Acest lucru se obţine prin integrarea conceptelor orientate obiect cu tehnologiile convenţionale ale bazelor de date.

Termenul de "orientare obiect" este folosit într-o varietate atît de mare de cazuri, încît chiar şi experţilor le vine greu să dea un echivalent exact al acestei denumiri într-un context bine determinat. Termenul apare în diferite domenii şi anume: limbaje de programare, baze de date, baze de cunoştinţe, inteligenţă artificială şi software.

La întrebarea: ce este de fapt un SGBD orientat obiect şi care sînt avantajele sale faţă de sistemele convenţionale nu poate fi dat nici un răspuns atîta timp cît nu se cunosc cei mai importanţi termeni care se folosesc aici: obiectul, clasa, ereditatea, metodele şi mesajele, fără a se cunoaşte modelul care a stat la baza sa.

Modelul orientat obiect

Orientarea obiect poate fi considerată ca o tehnică de organizare a unui sistem în vederea tratării obiectelor care apar în el. Conceptul de bază este principiul de încapsulare: datele care descriu o unitate logică împreună cu caracteristicile lor şi cu operaţiile definite asupra acestor date vor fi încapsulate într-un obiect. Datele pot fi citite şi modificate, asemănător cazurilor de tipuri de date abstracte, doar prin folosirea operaţiilor definite în acest scop.

Astfel, modificări ale reprezentării structurale a datelor unui obiect nu are efect direct asupra celorlalte obiecte.

Obiectele comunică schimbînd mesaje între ele. Un mesaj transmis unui obiect are ca efect executarea unei anumite operaţii asupra acestui obiect. Acest mecanism mai este denumit şi Message Passing. În cadrul limbajelor de programare convenţionale programatorul este răspunzător de alegerea corectă a operatorului. El identifică de pildă codul de program pentru adunarea a două numere complexe A=6+3i şi B=3-2i cu numele funcţiei "Complex Add", atunci cînd scrie: ComplexAdd(A,B). Dacă el doreşte să adune cifrele binare C=101101 şi D=10011, el trebuie să folosească funcţii corespunzătoare, şi anume BinaryAdd(C,D). Deci utilizatorul funcţiilor ComplexAdd şi BinaryAdd identifică odată cu numele funcţiei şi o parte a codului program, care va trebui executat, deci el decide cum va fi executată o operaţie. Modelul programării orientate obiect transferă responsabilitatea de la utilizatorul unei funcţii la proiectantul acelei funcţii.

Într-un sistem orientat obiect, programatorul poate scrie exemplele de mai sus în forma unor mesaje A.Add(B) şi C.Add(D), dacă operaţia Add a fost definită în prealabil atît pentru numere complexe, cît şi pentru cele binare. El va specifica doar ce trebuie executat. Care algortim se foloseşte la adunare, se va determina automat, în funcţie de obiectele A, B, C şi D în cauză, deoarece la încapsulare a fost stabilit cu exactitate care dintre operaţii se pot efectua cu datele respective. Deci în cazul A.Add(B) va fi executat automat algoritmul de adunare a numerelor complexe, iar în cazul C.Add(D) cel al adunării numerelor binare. Acest mecanism poartă denumirea de polimorfism.

Obiectele de acelaşi fel se pot clasifica în clase de obiecte. Clasele folosesc ordonării ontologice ale obiectelor care apar şi se prelucrează într-un sistem. Obiectele unei clase se numesc instanţe ale acelei clase. Astfel de exemplu numerele complexe A şi B de mai sus sînt instanţe ale clasei Complex Number, iar C şi D, instanţe ale clasei Binary Number.

Prin clasa unui obiect se determină atît reprezentarea internă a datelor, deci structura unui obiect, cît şi mulţimea operaţiilor definite asupra obiectului.

Pentru a obţine părţi modulare de sisteme, definiţiile de clase ar trebui să poată fi reutilizabile şi la alte definiţii de clase. Acest lucru se poate obţine prin conceptul eredităţii. Definiţiile unei clase pot fi derivate din alte definiţii de clase, deci o clasă nouă se construieşte pe baza unei clase existente, adăugînd acesteia noi definiţii de metode şi structuri interne de date. Procesul acesta poartă numele de specializare, deoarece clasele noi detaliază instanţele lor şi le descriu mai specific, decît clasele mai generale folosite la compunerea lor.

Aceste principii sînt comune tuturor sistemelor orientate obiect, cu toate că

detaliile sînt realizate şi definite în mod diferit. Orientarea obiect uşurează de multe ori, pe baza modelării naturale, înţelegerea sistemelor, a analizei lor, a proiectării, realizării şi a întreţinerii lor.

Sisteme de gestiune a bazelor de date orientate obiect

SGBD-urile orientate obiect se detaşează prin integrarea funcţionalităţii bazelor de date convenţionale cu concepte de modelare orientate obiect. Scopul lor primar este de a separa reprezentarea şi administrarea datelor de programele utilizator, care prelucrează aceste date, şi de a permite o utilizare consistentă a datelor de către mai multe programe utilizator, astfel încît programele să devină independente de reprezentarea internă a datelor. Totuşi responsabilitatea interpretării consistente şi a manipulării datelor rămîne în sarcina programelor utilizator.

Gradul înalt de complexitate al cîmpurilor aplicaţiei, cum ar fi, de ex., prelucrarea de documente puternic structurate, necesită transferul unei părţi a acestei responsabilităţi de la programele utilizator către datele structurate şi anume în SGBD. Cu alte cuvinte, programele utilizator nu trebuie doar să devină independente de reprezentarea internă a datelor, ci şi de semantica datelor, şi anume, de operaţiile ce se execută asupra acestor date. Doar astfel se poate garanta independenţa de aplicaţiile specifice a interpretării şi manipulării corecte a informaţiilor structurate complex, prin intermediul SGBD-urilor.

Adăugirea avută în vedere în cadrul proiectului VODAK pentru un SGBD deschis, orientat obiect, elimină următoarele îngrădiri ale sistemelor convenţionale de baze de date:

- reprezentarea structurată: obiectele structurate complex pot fi depuse integral în baza de date. Ele nu trebuiesc descompuse în părţi unitare plane, pentru a fi recompuse mai apoi de către programul utilizator. Acest lucru este posibil pe de o parte datorită construcţiilor tip predefinite cum ar fi Tuple, Set şi altele, iar pe de altă parte prin posibilitatea de a putea defini clase de obiecte oricît de complexe.
- prelucrarea datelor: programele utilizator sînt separate atît de reprezentarea internă a datelor cît şi, în mare parte, de semantica datelor.

Astfel, programele utilizator devin invariabile nu numai faţă de schimbarea formatului datelor, ci obţin în mare măsură şi o interpretare şi o manipulare consistentă a datelor conform semanticii lor, toate acestea fiind garantate de SGBD.

- model deschis, extensibil.

În comparaţie cu o serie dată de primitive de modelare, care stau la baza definirii unei "scheme" a unei baze de date într-un SGBD convenţional mulţimea primitivelor de modelare poate fi lărgită cu primitive de modelare specializate pe o anumită aplicaţie.

Astfel modelul de date al SGBD-urilor orientate obiect poate fi ajustat în funcţie de cerinţele unei aplicaţii.

În cadrul VODAK în colaborare cu alte proiecte din Institutul pentru Sisteme Integrate Informaţionale şi Publicistice, se execută asemenea ajustări şi extinderi de model în scopul modelării de documente hypertext, multimedia şi pentru integrarea unor baze de date eterogene deja existente.

În cadrul integrării unor concepte de modelare orientate obiect şi a funcţionalităţii SGBD-urilor convenţionale este necesară şi o anumită adaptare a unor funcţii fundamentale ale SGBD-ului.

Doar astfel se poate ajunge la o concordanţă optimă a serviciilor caracteristice ale unei baze de date, cum ar fi de exemplu managementul de tranzacţii, cu conceptele unui model orientat obiect.

Astfel în cadrul VODAK se exploatează de exemplu comutativitatea operaţiilor asupra obiectelor, pentru a obţine un paralelism mai ridicat în sistem. Acest fapt este important, deoarece structurile liber definite ale obiectelor unei aplicaţii pot deveni foarte mari şi din acest motiv pot exista multe accedări paralele asupra aceluiaşi obiect.

De aceea, reducerea la minim a unor asemenea accedări este o premisă importantă în obţinerea unor performanţe ridicate la un SGBD orientat obiect.

Deoarece o operaţie asupra unui obiect poate duce la apelarea altor operaţii asupra aceluiaşi obiect şi a altor obiecte, tranzacţia VODAK constă din mai multe subtranzacţii.

Această "imbricare" a tranzacţiilor are avantajul că fiecare operaţie se derulează ca o tranzacţie. Deci operaţia va fi executată complet sau sistemul va fi pus în mod automat în starea consistentă dinaintea executării operaţiei. Dacă există subtranzacţii care pot fi derulate cu succes, atunci rezultatele acestora devin instantaneu vizibile şi utilizabile de către alte tranzacţii, şi nu doar atunci cînd întregul proces de tranzacţii a fost încheiat.

Dacă o tranzacţie globală trebuie întreruptă dintr-un motiv anume, atunci subtranzacţiile care deja au fost derulate cu succes vor fi repuse în starea lor iniţială, cu ajutorul aşa-numitelor tranzacţii de compensare.

Deci datorită puterii mari de modelare şi a integrării conceptelor orientate obiect cu funcţionalitatea unei baze de date, SGBD-urile orientate obiect au în anumite aplicaţii avantaje mari faţă de sistemele convenţionale.

Prototipuri ale unor astfel de sisteme, pe lîngă cel din Institutul pentru Sisteme Integrate Informaţionale şi Publicistice, se experimentează în toată lumea, în laboratoare de cercetare pentru a dezvolta în continuare tehnologia bazelor de date orientate obiect pînă în faza în care vor putea fi prezentate pe piaţă.

Componente ale unor astfel de sisteme sînt implementate deja în unele produse fără ca acestea să dispună însă de toate caracteristicile de performanţă amintite.

(I.M.)

Blob, blob ...

(continuare din pag. 22)

Firma Data Acess din Friedrichsdorf oferă deja o soluţie. SGBD-ul "Dataflex", în versiunea 3.0 permite cîmpuri de tip BLOB cu o dimensiune maximă per cîmp de 16.384 octeţi. Dacă un fişier binar depăşeşte această dimensiune, atunci acesta poate fi divizat în mai multe cîmpuri. Teoretic, în acest mod, în fiecare tabelă a bazei de date, ar putea fi stocat un fişier binar avînd dimensiunea maximă de 16 milioane ori 16.000 octeţi. O mărire a lungimii unui cîmp BLOB nu se are însă în vedere la Data Acess.

Pentru tratarea unor cîmpuri de date mari, cum ar fi de pildă imaginile scanate, se oferă un program utilitar. Acesta preia afişarea imaginilor într-o fereastră, ca şi memorarea şi citirea datelor din fişier. Dataflex 3.0 rămîne răspunzător, în schimb, de manipularea datelor.

(R.M.)

Baze de date Client-Server

Care este diferenţa între un Sistem de Gestiune a Bazelor de Date (S.G.B.D) integrat şi un model Client-Server? Ce avantaje prezintă stocarea centralizată a datelor şi prelucrarea descentralizată a lor şi ce implicaţii au în acest caz tehnologiile moderne de realizare a reţelelor? În articolul care urmează vom încerca să răspundem acestor întrebări.

Pentru a putea decide care S.G.B.D. este mai indicat pentru un anume domeniu aplicativ, vor trebui lămurite mai întîi care sînt cerinţele de calitate care se pretind sistemului: cît de mare va deveni baza de date? în ce scopuri vor fi manipulate datele şi cărui cerc de utilizatori vor trebui să fie accesibile? Noţiuni ca redundanţă, integritate, consistenţă sau actualitate vor fi alte criterii hotărîtoare în cazul unei evaluări.

S.G.B.D.-uri relaţionale

S.G.B.D.-urile care pun în relaţie mai multe fişiere sînt denumite S.G.B.D.-uri relaţionale. Într-un S.G.B.D. relaţional fişierele sînt denumite tabele. Tabelele sînt înlănţuite prin intermediul unui cîmp comun, de exemplu numele. Să dăm un exemplu pentru clarificare: secţia de reclamă a unui supermagazin are nevoie de adresele clienţilor pentru a le putea expedia broşuri de reclamă. Secţia de creditare în schimb, are nevoie în primul rînd de numele clienţilor şi de datele referitoare la cont pentru a putea face înregistrările contabile. Dacă un client nu este capabil să returneze un credit acordat, atunci secţia de creditare va trebui să scrie o somaţie de plată şi s-o expedieze clientului. Pentru aceasta are nevoie de adresă. Prin intermediul cîmpului comun "nume" se poate face o legătură între tabela "adresă" şi "conturi". Într-un SGBD nerelaţional anumite date trebuie stocate în ambele fişiere, pentru a fi accesibile ambelor secţii. Lipsa relaţiei are ca dezavantaj faptul că datele trebuie stocate de mai multe ori. Acest fapt ridică anumite probleme pe care le vom descrie în continuare.

Date redundante

Se vorbeşte despre redundanţă atunci cînd o bază de date conţine date echivalente stocate de mai multe ori. Dacă, de exemplu, adresa domnului Moraru este memorată atît la secţia de desfacere cît şi la contabilitate şi la secţia de creditare, aceleaşi date sînt stocate de trei ori. Această metodă implică ocuparea triplă a spaţiului de memorare, şi nu este economică. Două din cele trei adrese stocate sînt în plus, deci sînt redundante. În cazul stocării centralizate a datelor se reduce redundanţa deoarece datele sînt memorate într-un singur loc, şi cu toate acestea sînt accesibile tuturor secţiilor.

Integritatea datelor

Dacă adresa domnului Moraru trebuie modificată, din cauza unei modificări de domiciliu, de ex., în cazul unei stocări descentralizate această adresă trebuie modificată la toate secţiile (de trei ori în cazul nostru). Integritatea datelor - se vorbeşte adesea şi despre consistenţa datelor - este periclitată dacă nu toate adresele sînt modificate cît mai rapid (dacă se poate simultan).

Actualitate

Pe actualitatea datelor se pune un preţ foarte mare. Căci la ce-i serveşte secţiei contabilitate faptul că secţia desfacere are fişierul de adrese la zi dacă somaţiile de plată proprii sînt returnate cu menţiunea "destinatar necunoscut"? De aici se poate vedea cît de strîns legate între ele sînt noţiunile redundanţă, integritate şi actualitate

Modelul Client - Server

Modelul Client - Server oferă o alternativă la modelul SGBD integrat. Într-un SGBD integrat baza de date propiu-zisă şi suprafaţa utilizator alcătuiesc un tot unitar. În baza de date - aşa-numitul Backend - sînt stocate fizic datele, în timp ce prin intermediul suprafeţei utilizator - aşa-numitul Frontend - datele sînt definite, manipulate şi controlate. Această unitate conceptională implică însă şi o unitate fizică, aceasta înseamnă că baza de date este instalată pe un singur PC. Ca urmare datele vor fi accesibile doar utilizatorului acelui PC.

O reţea, din contră, constă dintr-un calculator de comandă puternic - server-ul - şi mai multe staţii de lucru, aşa-numitele Workstations. Într-un SGBD care lucrează în reţea aceste două unităţi logice şi fizice sînt denumite Server şi Client. Astfel server-ul poate fi un PC puternic, de ex. un 386 sau 486, dar poate fi şi un minicalculator sau microcalculator. Clienţii sînt legaţi de server prin intermediul cartelelor de reţea şi al cablului şi nu trebuie să aibă aceeaşi configuraţie cu server-ul. În mod obişnuit sînt utilizate PC-uri 286 sau 386. Pe aceste PC-uri pot fi utilizate diferite pachete software, care însă să ofere o interfaţă unitară pentru comunicarea cu server-ul. Această interaţă este realizată de un limbaj de interogare standardizat, aşa-numitul Structured Query Language, cunoscut sub prescurtarea SQL. Posibilitatea de a utiliza pachete soft diferite permite o prelucrare şi/sau o reprezentare corespunzătoare. Astfel un program de gestiune a stocurilor poate uşura examinarea rezervei de articole din stoc. Un procesor de texte cu funcţie de scriere "scrisori-în-serie" va avea nevoie de adresele din baza de date şi va constitui astfel un frontend adecvat pentru secretariat. În fine un program de calcul tabelar cu funcţii grafice integrate va permite o reprezentare sub formă de diagrame a datelor.

Modelul client-server versus mainframe

Avantajele utilizării de PC-uri - şi deci a modelului client-server - comparativ cu modelul calculatoarelor mari (mainframe) sînt imediate. Calculatoarele mari împreună cu terminalele neinteligente alcătuiesc un sistem. Terminalele neinteligente sînt aparate de monitorizare, care nu dispun de inteligenţă proprie şi care reprezintă datele aşa cum le primesc de la calculatorul mare. O reprezentare pretenţioasă va implica o solicitare suplimentară a calculatorului care în acelaşi timp trebuie să se ocupe şi de gestiunea datelor. Situaţia se prezintă cu totul altfel în cazul utilizării reţelelor de PC-uri, în care sînt utilizate terminale inteligente, respectiv PC-uri, pentru a prelucra în continuare datele obţinute în urma interogării. Pe lîngă posibilitatea utilizării unor pachete soft diferite şi hardware-ul (capacitate hard-

disk, facilităţi grafice, putere procesor, etc.) poate fi adaptat necesităţilor concrete.

Reţele eterogene

Din fericire în cazul reţelelor de calculatoare moderne nu se mai impune trasarea unei linii de demarcaţie între familiile de calculatoare. Cuvîntul magic este reţele eterogene. Spre deosebire de o reţea omogenă, în care se pot cupla doar calculatoare de acelaşi tip (de ex. numai PC-uri compatibile IBM), într-o reţea eterogenă pot fi cuplate calculatoare din familii şi cu platforme diferite. De exemplu datele pot fi stocate pe un calculator mare şi suplimentar pot fi utilizate şi mediile de memorare ale mini- sau microcalculatoarelor. Operarea simplă a PC-urilor permite o manipulare a lor (gestiune şi reprezentare) pe staţia de lucru. Pentru aceasta, în funcţie de aplicaţie, pot fi utilizate atît calculatoare compatibile IBM (de ex. pentru calcul tabelar şi prelucrare de texte), cît şi Macintosh (de ex. pentru grafică) sau alte tipuri de calculatoare. În acest mod se obţine un efect de sinergie, prin utilizarea caracteristicilor complementare ale calculatoarelor individuale.

Prelucrarea tranzacţiilor

Deoarece în cazul unui SGBD client-server se gestionează un sistem complex de date distribuite, trebuie să se acorde o atenţie deosebită securităţii datelor. În centrul atenţiei stă în acest caz noţiunea de tranzacţie. O tranzacţie constă dintr-o succesiune de comenzi elementare care trebuie executate ca o singură operaţie. O cădere de sistem nu trebuie să ducă la întreruperea unei tranzacţii în timpul prelucrării ei. Dar cum se poate asigura acest lucru? Vom încerca să lămurim această situaţie prin exemplul următor:

Fie o tranzacţie care să fie alcătuită din comenzile: citeşte articole din tabela A, copiază aceste articole în tabela B şi şterge vechile articole din tabela A. Această procedură descrie deci o simplă mutare a articolelor din tabela A în tabela B. Dacă, de ex., comanda din mijloc nu este executată (de ex. din cauza unei erori de sintaxă), datele din tabela A nu vor mai fi copiate în tabela B şi cu toate acestea ele vor fi şterse din A. În cazul unei tranzacţii, în schimb procedura începe cu BEGIN WORK. Dacă nici una dintre instrucţiunile conţinute în tranzacţie nu produce mesaje de eroare şi nu există nici un motiv care să necesite o reluare a operaţiilor, atunci tranzacţia va fi încheiată cu COMMIT WORK. Abia în acest moment modificările sînt scrise în baza de date cu caracter permanent. Dacă în schimb în timpul tranzacţiei a apărut o eroare, cu comanda ROLLBACK WORK se poate reface vechea stare, fără pierderi de date.

Restaurarea datelor

Prin metoda de prelucrare a tranzacţiilor şi comanda ROLLBACK se asigură deci o consistenţă a datelor. Uneori se poate întîmpla, totuşi, ca o cădere de sistem să strice consistenţa datelor. Pentru a putea reface starea de dinaintea căderii (pentru aceasta se utilizează termenul englezesc Recovery) SGBD-ul ar trebui să realizeze două lucruri: să înregistreze un protocol al tranzacţiilor (Journal) şi să posede funcţia ROLLFORWARD.

Pentru aceasta este citit mai întîi ultimul backup şi este declarat apoi ca baza de date curentă. Cu ajutorul instrucţiunii ROLLFORWARD Bază_de_date toate tranzacţiile care fuseseră executate se execută din nou pentru noua bază de date (restaurată). În acest mod se reface din nou vechea stare a bazei de date.

Pe lîngă protocolul tranzacţiilor, care se referă la operaţiile care afectează întreaga bază de date, mai există şi aşanumitul protocol Audit. Acesta din urmă se referă la modificările survenite în cadrul unei tabele. Pentru fiecare tabelă poate fi executat deci un protocol audit şi pot fi efectuate astfel reconstrucţii ale unei tabele anume. Deoarece în cazul unui SGBD relaţional în cazul unei căderi de sistem, sau al unor instrucţiuni eronate, sînt afectate, de regulă, mai multe tabele înlănţuite, un protocol al tranzacţiilor este mai indicat şi mai confortabil. Din motive de complexitate un protocol audit este executat de regulă la un SGBD integrat, în timp ce la SGBD-urile client-server se utilizează, de regulă, protocolul tranzacţiilor.

Mecanisme de blocare

Prin Locking se înţelege un mecanism automat de blocare care devine activ în cazul accedării simultame a mai multor utilizatori asupra aceleiaşi colecţii de date. În cazul unui mediu multiuser (multiutilzator) se poate întîmpla ca mai mulţi utilizatori să acceadă simultan la aceeaşi tabelă. Operaţiile pure de citire nu pun probleme. Dacă în schimb utilizatorul Moraru modifică date într-o tabelă (şterge, editează sau inserează articole) şi în acelaşi timp utilizatorul Şulea face acelaşi lucru, atunci SGBD-ul nu poate decide care dintre modificări trebuie preluate. Mecanismul de blocare (locking) blochează execuţia operaţiunilor unui utilizator. Deoarece în cadrul conceptului client-server operaţiile sînt prelucrate ca tranzacţii, o blocare nu provoacă o întrerupere ci emite doar un mesaj din care să reiasă că tranzacţia n-a putut fi executată.

Nivele de blocare

Mecanismele de blocare pot fi poziţionate pe diferite nivele. Cea mai ieftină şi cea mai simplă soluţie o constituie blocarea unui întreg fişier atunci cînd doi utilizatori acced simultan la aceeaşi bază de date. Totuşi în acest

Setul de comenzi SQL(standardul ANSI-86 şi extensiile)

Limbajul de manipulare a datelor - DML
- BEGIN DECLARE SECTION
- CLOSE
- DECLARE CURSOR
- DELETE
- DELETE ... WHERE CURRENT OF
- DESCRIBE
- END DECLARE SECTION
- END-EXEC
- EXEC SQL
- EXECUTE
- EXECUTE IMMEDIATE
- FETCH
- Funcţia: AVG
- Funcţia: COUNT
- Funcţia: MAX
- Funcţia MIN
- Funcţia: SUM
- FROM
- GROUP BY
- HAVING
- INCLUDE
- INSERT
- INTERSECT
- MINUS
- OPEN
- ORDER BY
- Predicat: BETWEEN
- Predicat: EXISTS
- Predicat: IN
- Predicat: LIKE
- Predicat: NULL
- Predicat: SOME
- Predicat: ANY
- Predicat: ALL
- Predicat: operatori de comparaţie
- PREPARE
- SELECT
- SELECT ... INTO
- SUBSELECT
- UNION
- UPDATE
- UPDATE ... WHERE CURRENT OF
- WHENEVER
- WHERE

Limbajul de definire date - DDL
- ALTER TABLE
- COMMENT ON
- CREATE INDEX ON
- CREATE SYNONYM
- CREATE TABLE
- CREATE VIEW
- DROP INDEX
- DROP SYNONYM
- DROP TABLE
- DROP VIEW
- UPDATE STATISTICS

Limbajul de control al datelor - DCL
- COMMIT
- GRANT
- LOCK TABLE
- REVOKE
- ROLLBACK

caz se poate bloca prelucrarea diferitelor tabele care aparţin aceleiaşi baze de date chiar şi atunci cînd nu apar conflicte.În cazul unui număr mare de utilizatori aceasta ar putea conduce la o blocare permanentă. Utilizatorii trebuie să aştepte, în acest caz, mult timp pînă cînd baza redevine liberă. O astfel de soluţie nu este deci economicoasă.

O blocare la nivelul cîmpurilor individuale ar prezenta avantaje considerabile. Mai mulţi utilizatori pot prelucra, în acest caz, aceeaşi tabelă fără să se blocheze între ei atît timp cît nu editează acelaşi cîmp. Totuşi un astfel de mecanism de blocare este greu de realizat şi presupune o reducere a performanţelor (vitezei) de prelucrare, deoarece tot timpul cîmpurile unui fişier trebuie verificate pentru a nu fi accedate simultan.

Deadlock

Un aşa-numit deadlock poate să apară atunci cînd mai multe procese se blochează reciproc şi pînă la urmă nu mai poate fi executat nici unul. Acest lucru se poate întîmpla, de exemplu, atunci cînd utilizatorul Moraru deschide tabela A, modifică date în această tabelă şi vrea să copieze aceste date în tabela B, în timp ce utilizatorul Şulea tocmai a efectuat modificări în tabela B şi vrea să le copieze în tabela A. Operaţiile utilizatorilor Moraru şi Şulea sînt executate ca tranzacţii şi pot fi încheiate doar după terminarea tuturor prelucrărilor. Dar întrucît tabelele A şi B sînt blocate datorită accedării simultane, niciuna din cele două tranzacţii nu poate fi încheiată şi astfel se ajunge la un deadblock.

Un SGBD trebuie să observe apariţia unui astfel de deadlock şi să avertizeze utilizatorul asupra acestui lucru. Un deadlock poate fi evitat, de ex., dacă atunci cînd o tranzacţie nu poate fi executată într-un anumit interval de timp este întreruptă. În acest caz trebuie executat un rollback automat pentru a reface starea iniţială a sistemului.

Este uşor de înţeles astfel de ce noţiunea de tranzacţie este atît de strîns legată de cea de mecanisme de blocare şi ce influenţă au ele asupra calităţii securităţii datelor şi deci a integrităţii.

SQL ca limbaj de interogare standard

Posibilitatea de a implementa baze de date distribuite conform conceptului client-server, pe toate platformele hard, a impus necesitatea creării unui limbaj de interogare standard, a cărui sintaxă să fie aceeaşi pe toate calculatoarele. Structured Query Language, mai scurt SQL, a fost conceput la începutul anilor 80 ca limbaj de interogare pentru SGBD-ul IBM-ului DB2 disponibil pe calculatoare mari (mainframe). Ulterior American National Satndards Institut (ANSI) şi International Standards Organisation (ISO) au impus SQL ca limbaj de interogare standard pentru SGBD-urile de nivel ridicat. Practic toate SGBD-urile client-server dispun de o interfaţă SQL, al cărei set de comenzi corespunde cel puţin acestor norme şi care adesea este extins cu comenzi suplimentare. Despre Embedded SQL se vorbeşte atunci cînd setul de comenzi SQL este imbricat într-un limbaj de programare, ca de ex. Cobol, Fortran, PL/1 sau C. SQL constă în principiu din trei componente: limbajul de definire a datelor (Data Definition Language - DDL), limbajul de manipulare a datelor (Data Manipulation Language - DML) şi limbajul de control a datelor (Data Control Language - DCL).

Cu DDL se defineşte baza de date şi tabelele conţinute în ea. Tabelele sau cîmpurile pot fi deci inserate, şterse sau li se pot modifica atributele. De exemplu, tipul de cîmp char poate fi modificat în integer.

DML conţine, ca elemente de limbaj, comenzi pentru modificarea datelor. Datele pot fi şterse, inserate, editate, însumate cu ajutorul unei funcţii, poate fi calculată media artimetică sau maximul unui cîmp. Tot aici sînt incluse toate tipurile de sortări, selecţii şi grupări. Aşa-numitele instrucţiuni dinamice DML, care sînt utilizate în cadrul limbajelor de programare ca Embedded SQL, pot fi manipulate în timpul execuţiei programului sau li se pot varia argumentele. Deoarece aceste comenzi nu sînt încă standardizate, în funcţie de implementare, pot să difere mult de la un limbaj de programare la altul.

DCL este foarte important mai ales în cazul sistemelor multiutilizator, deoarece el controlează securitatea sistemului şi verifică drepturile de acces. În plus el permite o gestiune a tranzacţiilor şi în caz de necesitate poate executa o refacere a bazei (rollback).

În principiu SQL oferă deci o interfaţă unitară între backend şi diferitele front-end (programe de calcul tabelar, generatoare de rapoarte, prelucrare de texte etc.).

Deoarece odată cu dezvoltarea tehnologiei reţelelor, limitele dintre PC-uri, minicalculatoare şi calculatoare mai dispar din ce în ce mai mult, conceptul client-server ca soluţie integratoare cîştigă tot mai mult în însemnătate. SGBD-urile integrate rămîn interesante pentru o utilizare locală fără comunicaţii cu date externe, bazele de date gigant din lumea calculatoarelor mari se dovedesc a fi îmbătrînite şi a nu mai fi moderne; fără a mai vorbi de costurile ridicate pe care le implică întreţinerea unor astfel de sisteme.

Gupta SQL Base Server

SGBD-ul Gupta constă din mai multe componente: ca backend este utilizat SQL Base Server, care a fost realizat special pentru reţele locale LAN (Local Area Network). Din această cauză SQL Base Server suportă toate tipurile de reţele sub sistemele de operare DOS şi OS/2. Se află în pregătire şi o versiune Unix şi în versiunea 5.0 SQL Base este disponibil şi ca NLM (Netware Loadable Module) pentru Novell Netware 386. Pentru utilizarea în LAN, produsul a fost optimizat pentru OLTP (OnLine Transaction Processing) şi pentru pregătirea şi luarea de decizii (Decision Support). Aplicaţiile OLTP cer pe de o parte actualizări multiple, concurente ale datelor, pe de altă parte reacţii (răspunsuri) rapide. Aplicaţii clasice OLTP sînt, de ex., rezervarea de locuri la avion, sau aplicaţiile agenţiilor de asigurări. Cîteva caracteristici ne vor arăta cum reuşeşte să obţină Gupta performanţe ridicate:

- procesorul SQL preia sarcinile sistemului de operare prin funcţii proprii de Scheduling, Task Switching, Locking şi Disk Caching.
- pentru securitatea datelor se ţine un protocol combinat al activităţilor server-ului, în cazul în care mai mulţi utilizatori îşi termină tranzacţiile aproape simultan cu COMMIT.
- buffer-ul de date permite acces multiutilizator, astfel încît un utilizator poate accede şi la datele utilizate simultan de altcineva.
- inserări şi ştergeri multithreaded de linii din tabele.

Ca suport pentru luarea de decizii Gupta susţine programe de calcul tabelar, interogări de baze de date şi crearea de rapoarte. O funcţie specială browse permite "răsfoirea" simultană în mulţimi de date diferite. Articolele pot fi actualizate oricînd în succesiunea dorită.

Ca sistem de dezvoltare grafic pentru aplicaţiile frontend este disponibil SQL Windows, care poate fi implementat sub Windows 3.0 pentru MS-DOS sau sub Presentation Manager pentru OS/2. Software-ul de mangement a bazei de date Quest permite utilizatorului final să acceadă la date, să interogheze o bază de date SQL şi să obţină rapoarte prin intermediul unei suprafeţe utilizator grafice.

Informix SE şi Online

Ca backend Informix oferă două server-e de baze de date: System-Engine (SE) şi Online. Ambele respectă standardul de interfaţă ANSI-SQL/89 şi pot lucra astfel cu toate programele frontend care respectă acest standard. Informix oferă următoarele programe utilitare: I-SQL, 4GL, 4 GL-Forms, Menus, ESQL/C, ESQL/Cobol şi Storm. Prin intermediul componentei integrate Datalink, la date pot accede şi programele Wingz (calcul tabelar şi grafică de prezentare) şi SmartWare (procesor de texte).

Informix SE şi Informix-Online au fost concepute pentru domenii aplicative şi pieţe diferite. SE atinge cele mai bune performanţe la un număr mic pînă la mediu de utilizatori. Efortul de instalare şi necesarul de memorie sînt mai reduse, gestiunea şi dimensiunea bazei de date sînt adaptate unui număr mai mic de utilizatori.

OnLine este SGBD-ul uriaş care îşi relevă avantajele în cazul operării multiutilizator. Caracteristici semnificative sînt gestiunea unor volume mari de date şi posibilitatea prelucrării distribuite (pot fi utilizate simultan mai multe baze de date). Pentru sporirea securităţii, OnLine permite oglindirea unor părţi ale bazei de date, sau a întregii baze. Aceasta înseamnă că în permanenţă datele sînt memorate pe două suporturi distincte.

Informix OnLine este singurul SGBD, din cele prezentate în acest articol, care oferă şi suport multimedia. El poate memora informaţii sub formă de text, grafică, voce. Fiecare cîmp poate avea dimensiunea de pînă la 2 GB. Cu produsul suplimentar OnLine/Optical, datele pot fi stocate pe un disc optic.

SE şi OnLine suportă baze de date distribuite, aceasta înseamnă că mai multe tabele ale aceleiaşi baze de date pot fi stocate pe medii de memorare şi server-e diferite. Acest lucru permite o mai bună utilizare a resurselor într-o reţea locală.

Ingres SQL - Server

Caracteristicile principale ale SGBD-ului client-server Ingres sînt performanţe optime pentru OLTP (OnLine Transaction Processing) şi suport pentru lucrul 24 ore din 24. Trei elemente fac din Ingres un SGBD inteligent: pe lîngă o gestiune simplă a datelor memorate (Data Management) mai sînt accesibile două facilităţi.

Gestiunea centralizată a regulilor unei întreprinderi prin aşa-numitul Knowledge Management. Pînă acum aceste reguli trebuiau codificate în fiecare program de aplicaţie. Prin Knowledge Management oricît de complexe ar fi regulile ele pot fi gestionate central şi deci trebuie definite o singură dată. Avantajele sînt următoarele:

- consistenţa datelor este asigurată
- programele utilizator devin mai simple şi mai lizibile
- întreţinerea regulilor este mai simplă
- administratorul bazei de date dispune de noi posibilităţi de acordare şi revocare de drepturi de acces pentru utilizatori, grupe de utilizatori şi aplicaţii.
- necesarul de resurse al utilizatorilor individuali poate fi supravegheat şi gestionat central de administratorul sistemului.

Interogările utilizator care suprasolicită harware-ul existent pot fi refuzate sau amînate. Performanţele întregului sistem pot fi astfel mai uşor calculate.

Object Management este a doua facilitate care încearcă să facă din Ingres un SGBD inteligent. El permite gestiunea unor obiecte complexe, cum ar fi de exemplu coordonatele sau obiectele geografice. Pînă acum obiectele trebuiau definite în structuri de date simple. Object Management permite definirea unor noi tipuri de date, operatori şi funcţii care să fie gestionate la fel de rapid ca şi tipurile de date standard.

Ingres oferă unelte interactive pentru suportul unor suprafeţe utilizator grafice (GUI), ca de exemplu pentru Windows 3.0. Cu ajutorul lui Ingres-Windows 4GL se pot crea baze de date într-un mediu de dezvoltare orientat obiect de generaţia a 4-a.

Pentru generarea automată de aplicaţii pentru terminale orientate caracter poate fi utilizat Ingres-Vision.

Oracle SQL-Server

OSE sau Open Software Environment este cuvîntul magic Oracle. Modelul OSE constă din şapte nivele:

1. Producători hardware: orice utilizator important trebuie să poată implementa un nou SGBD în mediul hard eterogen de care dispune. Oracle suportă hardware-ul tuturor producătorilor importanţi, ca de ex., IBM, Nixdorf, Siemens, DEC, HP, Bull, Sun şi Apple.

2. Arhitecturi hardware: IBM - Mainframe, sisteme multiprocesor VAX, staţii de lucru RISC produse de Nixdorf sau arhitecturi SPARC produse de SUN sînt doar cîteva exemple din multitudinea de platforme existente azi în lumea calculatoarelor şi pe care Oracle poate funcţiona. OSE este de asemenea deschis pentru tehnologiile moderne din industria tehnicii de calcul cum ar fi multiprocesare dedicată sau simetrică, procesare paralelă, arhitecturi multicomputer sau diferitele realizări LAN.

3. Sisteme de operare: Această rubrică s-ar fi putut intitula "de la PC pînă la mainframe". MVS, VM, VSE pentru mainframe-uri IBM, System 6150 sub AIX, Siemens BS 2000, DEC/VAX cu VMS şi Ultrix, peste 40 de derivate Unix, Macintosh, MS DOS şi OS/2 sînt doar o parte din sistemele de operare suportate.

4. Standarde industriale. De fapt n-ar trebui să existe decît un standard, realitatea însă este alta. Pe lîngă interfaţa standard SQL şi compatibilitatea cu DB2 al lui IBM, Oracle oferă interfeţe şi spre limbajele de programare, baze de date, aplicaţii şi suprafeţe grafice, utilitare frontend şi interfeţe de reţea şi de comunicaţie. Implementarea SGBD-ului Oracle în mediul sistem propriu permite utilizatorului să păstreze mediul obişnuit de lucru. Cîteva dintre suprafeţele utilizator posibile sînt enumerate în continuare: OSF/Motif, Open Look, Presentation Manager, Windows, DEC

Windows, suprafaţa utilizator Macintosh.

5. Procesare cooperativă: acesta este sloganul sub care Oracle integrează datele de bază situate în diferite locuri pe calculatoare provenind de la producători diferiţi. În legătură cu arhitectura client-server, Oracle SQL Star comunică prin intermediul unor protocoale diferite cu lumea eterogenă a sistemelor de calcul. Colecţiile de date de pe PC-uri, mainframe-uri sau staţii de lucru sînt gestionate central de către administrator, utilizatorul nu realizează de unde îşi extrage datele.

6. Resurse de informaţii: Bazele de date centrale, cum sînt bazele de date relaţionale ale IBM-ului, DB2 şi SQL/DS pot fi interogate prin intermediul lui Oracle SQL Connect din programe de aplicaţie care rulează pe un DEC/VAX sau pe un calculator Unix.

7. Modelele de integrare. Sistemul eterogen constituit din diferite platforme hard, S.O., reţele, baze de date este integrat într-un tot omogen.

Sybase SQL - Server

Arhitectura client-server Sybase constă din două familii de produse:

- SQL-Server preia gestiunea datelor şi controlul tranzacţiilor independent de aplicaţiile client şi interfaţa utilizator
- SQL - Toolset cuprinde un set de instrumente prietenoase pentru dezvoltarea şi operarea aplicaţiilor.

Integritatea datelor şi performanţe ridicate, aceasta este ceea ce Sybase doreşte să realizeze cu SQL-Server. Pentru aceasta sînt puse la dispoziţie mai multe funcţii. Aşa-numitele Stored Procedures (obiecte create şi compilate cu Transact SQL) împreună cu aşanumiţii triggers (proceduri care sînt lansate automat în execuţie atunci cînd cîmpurile respectă anumite condiţii) sînt utilizaţi pentru asigurarea integrităţi datelor. Regulile valabile pentru întreaga întreprindere (business rules) sînt stocate central, pe server, şi stau la dispoziţia tuturor clienţilor. Acest concept permite înlăturarea codului redundant, uşurează întreţinerea şi asigură controlul central pentru protecţia datelor întreprinderii.

Pentru atingerea unor performanţe ridicate este răspunzătoare arhitectura multi-threaded pe care se bazează SQL-Server. Ea cuprinde nucleul sistemului şi un SQL-Task-Manager care preia funcţiile multiutilizator cum ar fi: scheduling, task switching, disk caching şi locking. SGBD-ul este capabil ca în doar 1 MByte de memorie suplimentară să gestioneze 30 de utilizatori.

Prin intermediul aşa-numitei arhitecturi Virtual-Server sînt utilizate la capacitate toate procesoarele dintr-un sistem multiproceosr.

Cu Open Server se încearcă utilizarea Sybase şi într-un sistem eterogen. El permite atît clienţilor cît şi server-ului să comunice cu SGBD-uri străine, baze de date nerelaţionale, aplicaţii deja existente, sisteme în timp real, etc.

XDB-SQL-Server

XDB-SQL-Server este singurul SGBD prezentat aici, care în cea mai redusă formă a sa poate rula şi pe un calculator cu doar 640 KB de memorie. Produsul bazat pe o arhitectură client-server este complet. El dispune de funcţii GRANT/REVOKE pentru acordarea/revocarea de drepturi de acces, COMMIT, ROLLBACK pentru prelucrarea tranzacţiilor de mecanisme de blocare ca şi de compatibilitate SQL deplină cu ANSI/ISO-DB2 şi dialectele SAA-SQL.

Există două versiuni XDB-Sever: o versiune MS-DOS, pentru care serverul are nevoie de protocolul Net BIOS şi suportă pînă la 16 MB de memorie extinsă şi o versiune OS/2 care are nevoie de cel puţin 2 MB memorie şi de protocolul Net BIOS sau Named Pipes. Server-ul şi staţiile de lucru pot funcţiona sub sisteme de operare diferite.

Paleta de produse XDB mai conţine pe lîngă XDB-SQL-Server încă alte trei produse:

- XDB-SQL PLUS este un pachet single-user care conţine XDB-SQL-Engine şi SGDB-ul (DBMS), inclusiv Interactive SQL, Data Entry, Import/Export şi Data Dictionary. Acest pachet mai conţine şi diferite utilitare, cum ar fi un generator de măşti (forme), un generator de rapoarte şi Procedural Language ca limbaj de programare.
- XDB-WORKBENCH permite crearea unor aplicaţii DB2 pentru mainframe-uri IBM şi PC. Ca mediu de dezvoltare este utilizat limbajul Cobol.
- XDB-Tools sînt instrumente de dezvoltare pentru crearea de aplicaţii şi versiuni runtime.
- XDB-EXT este o versiune single-user extinsă, care poate gestiona pînă la 16 MB de memorie extinsă.
- XDB-C este o interfaţă şi o bibliotecă de utilitare pentru programe C care conţine un precompilator C, Embedded SQL şi API (Application Programming Interface) pentru comunicarea cu SQL Engine.
- XDB-Windows este un kit de dezvoltare pentru aplicaţii SQL sub Windows 3.0.

Novell Netware SQL

Şi Novell oferă o serie de produse care ca şi componente ale unui sistem de operare multiutilizator permit crearea unor baze de date client-server puternice. Ca backends sau SQL-Engines pot fi utilizate fie Netware Btrieve fie Netware SQL care în versiunea pentru AT 286 Netware 2.x pot fi cuplate static ca şi VAP (Value Added Process) respectiv în versiunea 386 Netware 3.11 pot fi cuplate dinamic ca şi NLM (Netware Loadable Module). Btrieve este un SGBD care a fost creat special pentru un File Handling rapid. Dintr-un limbaj de programare prin intermediul unor proceduri de sistem se poate accede secvenţial la datele Btrieve.

Netware SQL este un SGBD cu interfaţă SQL standard prin care se poate comunica cu diferitele front-end. Pentru proiectanţi, Netware SQL oferă o interfaţă compatibilă atît cu ANSI/SQL cît şi cu specificaţiile IBM SAA-SQL. Netware SQL poate fi utilizat în mai multe moduri:

- programele aplicative ca Lotus 1-2-3 pot accede direct la datele de pe backend prin intermediul Netware.
- Xtrieve PLUS al lui Novell este de asemenea un frontend bun dacă utilizatorul vrea să acceadă la un produs deja integrat în Netware şi nu vrea să cumpere alte pachete soft. Xtrieve dispune de un sistem de interogare condus prin meniuri (Query Manager) care permite şi generarea de rapoarte.
- Pot fi create aplicaţii proprii utilizînd instrumente de dezvoltare. Un astfel de instrument este RaSQL al firmei Communication Horizon Clipper.
- Proiectanţii pot scrie aplicaţii proprii utilizînd un limbaj de programare standard şi interfaţa utilizator XQL.

(R.M.)

Produs	SQLBase Server	Standard Engine 5.0	OnLine 5.0	Ingres Database	ORACLE Server	SYBASE SQL Server	XDB Server
Producător:	Gupta Technologies	Informix Software	Informix Software	Ingres	Oracle	Sybase	XDB Systems Inc.
Preţuri:							
Modul de bază	2.995 - 29.995 DM+ 14% TVA	la cerere	la cerere	la cerere	nec.	la cerere	peste 3900 DM
Extensii	nec.	Standard Engine (SE)	XA/TP,Optical OnLine	la cerere	nec.	la cerere	Precomp. MS-C 6.0,Cobol
Licenţă (Runtime)	1200DM+14%	la cerere	la cerere	la cerere	nec.	la cerere	la cerere
Date tehnice:							
Dimens. program în KB	600	700	1130	~ 50MB(Unix)	nec.	Server: 1500	7 MByte
Memoria rec. în MB	4 - 8	2	8	min. 8	nec.	16, (min.8)	peste 2
Versiune Windows 3.0	x	instr. suportă Windows 3.0	instr. suportă Windows 3.0	da,Ingres/ Windows4GL	nec.	ca Frontend	x
Expanded Memory	x	x	-	-	nec.	x	x
Extended Memory	-	x	x	x	-	x	-
Suport coprocesor	-	x	x	x	-	x	-
Sisteme de operare suportate:							
MS-DOS	x	x	-	da:Tools,ABF, Vision,ESQL, Win. 4GL,Net	x	numai Client	x
S.O. Macintosh	-	x	x	-	x	numai client	-
S.O. Next	-	x	planificat	-	x	x	-
Derivate UNIX	SUN-OS	x	x	x	>50 var. UNIX	x	încă nu
OS/2	x	x	-	x	x	x	x
DEC VMS	Connectivity şi NetWare 386	-	-	x	x	x	-
Reţele suportate:							
Novell Netware 2.1	x	File-Sharing	-	x	x	-	x
Novell Netware 3.1	x	-	x	x	x	x	x
LAN-Manager	x	-	-	x	x	x	x
IBM PC-LAN	x	File-Sharing	File-Sharing	-	x	x	x
Banyan VINES 4.0	x	File-Sharing	File-Sharing	în pregtire	x	x	x
Altele	nec.	AT & T StarGROUP	AT & T StarGROUP	nec.	nec.	nec.	nec.
Protocoale suportate:							
Token Ring	x	x	x	fnc. de reţea	nec.	x	fnc. de reţea
CSMA/CD	x	x	x	fnc. de reţea	nec.	-	fnc. de reţea
Arcnet	x	x	x	fnc. de reţea	nec.	-	fnc. de reţea
Ethernet	x	x	x	fnc. de reţea	nec.	x	fnc. de reţea
StarLAN	x	x	x	fnc. de reţea	nec.	x	fnc. de reţea
TCP/IP	x	x	x	x	x	x	da
IOS TP4	x	-	-	fnc. de reţea	-	-	fnc. de reţea
IBM NetBEUI/DLC	x	-	-	fnc. de reţea	x	-	fnc. de reţea
Altele	nec.	nec.	nec.	nec.	SPX/IPX,DECnet,APPC, Named Pipes,şi altele	nec.	Netbios, SPX, Named Pipes, APPC, Win. (loc.)
Volumul bazelor de date:							
Nr. max. tabele	nelimitat	nelimitat	14 milioane	nelimitat	nelimitat	32767	nelimitat
Nr. tabele desc. simultan	nelimitat	funcţie de S.O.	32000	nelimitat	nelimitat	configurabil	funcţie de S.O.
Nr. de articole per tabelă	nelimitat	2,1 miliarde	4,3 miliarde	nelimitat	nelimitat	2 miliarde	2 miliarde
Nr. cîmpuri per articol	254	32767	32767	300	255	250	400
Lungime cîmp. text în octei	254	32767	2 MB	2 KB	255	2 GB	4056
Nr. max. tabele înlnuite	nelimitat	funcie de S.O.	32000	30	oricîte	16	360
Cîmpuri:							
Alfanumeric	x	x	x	x	x	x	x
Numeric	x	x	x	x	x	x	x

Produs	SQLBase Server	Standard Engine 5.0	OnLine 5.0	Ingres Database	ORACLE Server	SYBASE SQL Server	XDB Server
Nr. zecimale	15	pîn la 32	pîn la 32	definibil	41	definibil	configurabil
Dat	x	x	x	x	x	x	configurabil
Timp	x	x	x	x	x	x	configurabil
Memo	x	x	x	-	x	x	x
Cîmpuri definite de utilizator	-	-	-	prin Object Management	-	x	-
Cuplare de imagini/grafic	x	-	BLOB	x	-	x	-
Calcule trigger	-	x	x	prin Knowledge Management	-	x	
Gestiune date:							
Ver. aut. a plauzabilităţii	x	x	x	x	x	x	x
Verificare duplicare articol	x	x	x	programabil	x	x	x
Cîmp obligatoriu	x	x	x	x	x	x	x
Conversie automată majuscule/minuscule	-	comand	comand	x	x	în instr. SELECT	x
Nr. de formate dat	definibil	oricîte	oricîte	oricîte	oricîte	nec.	5
Numerotare automată articole	-	x	x	x	x	la Stored Procedure	x
Preluare date din articolul anterior	-	x	x	x	x	x	x
Reorganizare automată/ Reindexare	-	comand	comand	x	x	x	x
Import/Export:							
Nr. formate import/export	7	1	1	nec.	nec.	2	x
DIF	x	prin Wingz	prin Wingz	nec.	nec.	-	x
dBase III	x	prin Wingz	prin Wingz	nec.	numai import	-	x
dBase III Plus	x	prin Wingz	prin Wingz	nec.	numai import	-	x
dBase IV	x	prin Wingz	prin Wingz	nec.	nec.	-	x
Lotus	-	prin Wingz	prin Wingz	nec.	numai Import	-	x
Framework	-	prin Wingz	prin Wingz	nec.	nec.	-	nec.
Excel	-	prin Wingz	prin Wingz	nec.	nec.	-	nec.
Quattro	-	prin Wingz	prin Wingz	nec.	nec.	-	nec.
ASCII	x	x	x	x	x	x	fixed ASCII, free ASCII
DB2	x	cu conv. ASCII	OpenView	prin Gateway	nec.	-	DBMAUI, DSNTIAUL
Format definit de utilizator	-	x	x	x	x	x	-
Altele	nec.	x	x	nec.	nec.	binar	Wordperfect: Mail merge files, Multiplan: SYLK
Întreţinere date:							
Ştergere simultană a mai multor articole	x	x	x	x	x	x	comandă SQL
Actualizare simultană a mai multor articole	x	x	x	x	x	x	comandă SQL
Recalculare automată	x	x	x	x	x	prin Trigger	nec.
Înlocuire automată	x	x	x	x	x	prin Trigger	nec.
Altele	nec.	nec.	nec.	nec.	nec.	nec.	Fetch Privious în ESQL, ISQL, Proceduri
Sortare date:							
În ordine crescătoare	x	x	x	x	x	x	max. 64 criterii
În ordine descrescătoare	x	x	x	x	x	x	max. 64 criterii
Alfanumeric	x	x	x	x	x	x	la alegere
Numeric	x	x	x	x	x	x	la alegere
Sortarea tuturor cîmpurilor	x	x	x	x	x	x	la alegere
Sortare doar a cheilor	x	x	x	x	x	nec.	la alegere
Nr. max. cîmp. sortate sim.	nelimitat	nr. cîmpurilor	nr. cîmpurilor	300	oricîte	16	64

Produs	SQLBase Server	Standard Engine 5.0	OnLine 5.0	Ingres Database	ORACLE Server	SYBASE SQL Server	XDB Server
Indexare date:							
Nr. max. cîmpuri indexabile	nelimitat	nelimitat	96	300	oricîte	16	400
Nr. max. fişiere index deschise simultan	nelimitat	nelimitat	nelimitat	nelimitat	oricîte	16	funcţie de S.O.
Act. aut. a fişierelor index	x	x	x	x	x	x	x
Indexare aut. la import date	x	x	x	x	x	x	x
Altele	nec.	nec.	nec.	nec.	nec.	nec.	Clustered Index: 1 per tabel
Chei şi indexare:							
Prime key	x	x	x	x	x	x	x
Foreign key	x	x	x	prin Knowledge Management	x	x	x
Integritate referenţială	x	x	x	prin Knowledge Management	x	x	x
Index	x	x	x	x	x	x	simplu şi unic
Query Optimizer	x	x	x	x	x	x	x
Altele	nec.	nec.	nec.	nec.	nec.	nec.	Compound Key: max. 20 comp.
Căutare date:							
Utilizare comenzi SQL	x	x	x	x	x	x	în Interactive SQL,Proceduri, Embedded SQL
Depistarea tuturor art. care satisfac condiţia de căutare	x	x	x	x	x	x	
Cupare logică căutări	x	x	x	x	x	x	x
Căutare fonetică	-	x	x	-	x	x	-
Căutare în cîmpuri memo	x	x	x	-	-	x	N/A
Generator de cereri	x	Utilitarul I-SQL	Util. dbaccess	x	x	x	-
Memorarea cererilor	x	x	x	x	x	x	x
Rapoarte:							
Rapoarte standard aut.	x	I-SQL,4GL	I-SQL,4GL	x	x	x	prin dialog şi ca procedură batch
Rapoarte grafice	x	Storm,Wingz	Storm,Wingz	-	-	alţi ofertanţi	produse străine
Memorare rapoarte	x	x	x	x	x	x	x
Formulare raport	x	I-SQL,Storm	I-SQL,Storm	x	x	x	x
Macrouri:		I-SQL,Storm, Wingz,4GL	I-SQL,Storm, Wingz,4GL				
Editor de macrouri	-	x	x	4GL	nec.	-	x
Înregistrare automată	-	x	x	4GL	nec.	-	x
Macro la o apăsare de tastă	-	x	x	4GL	nec.	-	x
Limbaj de macrouri propriu	-	x	x	4GL	nec.	-	SQL cu extensii
Protecţie date:							
Parolă program	x	x	x	x	x	x	x
Parolă fişier	x	-	-	x	-	x	autorizare (DBA)
Parolă tabel	-	-	-	x	-	-	autorizare (DBA)
Parolă cîmp	-	-	-	x	-	-	-
Protecţie automată date	x	la nivel de S.O.	x	x	x	-	meniu fiş. LOG
Backup automat	x	la nivel de S.O.	x	x	x	-	meniu fiş. LOG
Criptare date	x	implicit	implicit	-	-	-	-
Funcţii Recovery:							
Re-Index	x	x	x	x	x	-	x
Funcţie Undo	x	-	-	x	x	-	x
Funcţie Redo	x	-	-	x	x	-	x
Rollback	x	x	x	x	x	x	x
Rollforward	x	x	x	x	x	x	x
Fast Recovery	x	-	x	x	x	x	x

Produs	SQLBase Server	Standard Engine 5.0	OnLine 5.0	Ingres Database	ORACLE Server	SYBASE SQL Server	XDB Server
Prelucrare tranzacţii	x	x	x	x	x	x	da/nu la rulare
Tranzacţii imbricate	x	-	-	-	-	x	-
Verificări la nivel cîmp	x	x	x	x	x	x	Range/Defaults/ Values
Verificare structurală	x	x	x	x	x	x	ver. tip şi conv.
Verificare procedurală	-	x	x	x	x	x	via SQL în gen. de formulare
Cardinalitate	x	x	x	x	x	nec.	nec.
Commit în mai multe trepte	-	-	-	Commit 2 faze	x	x	-
Sincronizare	-	funcie de SGF	x	x	x	x	-
Moduri de blocare recunoscute:							
Blocare bază de date	nec.	x	x	x	x	-	prot. prin parolă
Blocare tabelă	nec.	x	x	x	x	x	Exclusive şi Share Intentional Locks
Blocare articol	nec.	x	x	x	x	-	x
Blocare multiplă articole	nec.	x	x	x	x	-	x
Dead-Locks	x	atenuare prin Nowait	recunoaştere şi rezolvare	recunoaştere şi tratare	x	sînt depistate de server	rec. automată la Autocommit
Altele	Page-Level	Page-Level	Page-Level	citire fără blocare	-	blocare pag.	nec.
Concept de operare:							
Windows 3.0	Quest, SQLWindows	prin instr. apl.	prin instr. apl.	x	x	suport	versiune Windows
Suport mouse	x	prin instr. apl.	prin instr. apl.	x	x	x	nec.
Tehnica ferestrelor	x	prin instr. apl.	prin instr. apl.	x	x	x	la DOS şi OS/2
Lucrul condus de meniuri	x	prin instr. apl.	prin instr. apl.	x	x	x	la DOS şi OS/2
Programare:							
Limbaj de prog. propriu	x	4GL	4GL	4GL	PL/SQL	x	XDB PRO
Nr. limb. prog. suportate	nec.	5 + de la teri	5 + de la teri	6	nec.	6	3
Interfaţă C	x	x	x	x	x	x	MS-C6.0, API Interface şi Embedded SQL-C
Interfaţă Cobol	x	x	x	x	x	x	Realia Cobol, Micro Focus Work-bench, Embed-ded SQL Cobol dinamic şi static
Interfaţă Basic	x	-	-	x	-	x	-
Interfaţă Ada	-	sub Unix	sub Unix	x	-	x	-
Interfaţă Pascal	x	-	-	x	-	x	-
Interfaţă Fortran	-	sub Unix	sub Unix	x	x	x	-
Alte interfeţe	interfaţă de prog. generică	interfaţă CALL	interfaţă CALL	-	-	nec.	XDB-Link la IBM Host DB2
Limbaj 4GL integrat	x	4GL	4GL	x	x	pentru Stored Procedures	x
Compilator integrat	x	compilator 4GL	compilator 4GL	x	-	x	pentru formulare şi proceduri
Versiune Runtime	-	x	x	x	-	de la terţi	la cerere
Instrumente grafice	SQLWindows, Quest, Chart Builder	Wingz	Wingz	x	x		produse străine
Generatoare:							
Generator de aplicaţii	x	4GL	4GL	x	x	x	în gen. de măşti
Generator de măşti	x	4GL Forms	4GL Forms	x	x	x	x
Generator de meniuri	x	Menus	Menus	x	x	x	x
Generator de rapoarte	x	x	x	x	x	x	x
Generator de etichete	x	x	x	x	x	-	nec.

Piaţa românească

a) Firma b) Telefon	a) Model b) Producător c) Procesor d) Preţ informativ	a) Tip carcasă b) Prod. procesor c) Tact(MHz) d) Waitstates	a) RAM stand. (MB) b) RAM max. (MB)	a) Harddisk standard (MB) b) Producător c) Timp de acces (ms)/Interleave d) Harddisk max. (MB)	a) Interfaţă/ Tip controller b) Nr. floppy 5,25" c) Nr. floppy 3,5"	a) Standard grafic b) Rezoluţia (pixeli) c) Nr. max. culori d) Monitor (inch)	a) Nr. interfeţe paralele/seriale/mouse b) Arhitectură bus c) Sloturi 8/16/32/64 biţi d) Din care libere 8/16/32/64	a) Sistem de operare b) Manuale c) Garanţie d) Termen livrare
PC-uri AT286								
a) CSI Ltd. b) 90/12.10.45	a) TCR b) - c) 80286 d) 252.485 lei	a) - b) Intel c) 12 MHz d) -	a) 1 MB b) -	a) 40 MB (+ 75.175 lei) b) - c) 23 ms/ - d) -	a) IDE b) 1 c) 1	a) MDA b) - c) - d) 14"	a) - b) - c) - d) -	a) - b) da c) 18 luni d) -
a) Elinco S.A. b) 90/12.20.30	a) Networks b) U.S.A. c) 80286 d) 375.000 lei	a) Baby b) Harris c) 20 MHz d) 0	a) 1 MB b) 16 MB	a) 40 MB b) Seagate c) 20 ms/ 1:3 d) oricît	a) IDE b) 1 c) 1	a) SVGA 512K b) 1024 x 768 c) 16 d) 14"	a) 1/2/0 b) ISA c) 2/5/0/0 d) 1/4/0/0	a) MS-DOS 5.0 b) da c) 12 luni d) 30 zile
a) System Plus SRL b) 90/79.13.91	a) A-Tronic b) - c) 80286 d) 350.322 lei	a) Slimline b) - c) 16 MHz d) -	a) 2 MB b) -	a) 80 MB b) - c) - d) -	a) IDE b) 1 c) 1	a) VGA 256K b) - c) monocrom (454.224 lei color) d) 14"	a) - b) - c) - d) -	a) - b) - c) - d) -
PC-uri AT386SX								
a) CSI Ltd. b) 90/12.10.45	a) TCR b) - c) 80386SX d) 363.899 lei	a) - b) Intel c) 20 MHz d) -	a) 1 MB b) -	a) 80 MB (+ 131.435 lei) b) - c) 19 ms/ - d) -	a) IDE b) 1 c) 1	a) MDA b) - c) - d) 14"	a) - b) - c) - d) -	a) - b) da c) 18 luni d) -
a) Elinco S.A. b) 90/12.20.30	a) Ares b) U.S.A c) 80386SX d) 540.000 lei	a) Baby b) AMD c) 25 MHz d) 0	a) 2 MB b) 16 MB	a) 52 MB b) Quantum c) 14 ms/ 1:3 d) oricît	a) IDE b) 1 c) 1	a) SVGA 512K b) 1024 x 768 c) 16 d) 14"	a) 1/2/0 b) ISA c) 1/5/1/0 d) 1/3/1/0	a) MS-DOS 5.0 b) da c) 18 luni d) 30 zile
a) MedComp DHS b) 90/12.30.37	a) Copam b) - c) 80386SX d) 2.555 DM	a) - b) Intel c) 8/20 MHz d) -	a) 2 MB b) 8 MB	a) 40 MB b) - c) 19 ms/ - d) -	a) IDE b) 1 c) 1	a) VGA 512K b) 1024 x 768 c) - d) 14"	a) 1/2/0 b) - c) - d) -	a) MS-DOS 5.0; Windows 3.0 b) -, c) - d) -
a) System Plus SRL b) 90/79.13.91	a) Iverson b) U.S.A c) 80386SX d) 620.000 lei	a) - b) - c) 16 MHz d) -	a) 2 MB b) -	a) 80 MB b) - c) -/ - d) -	a) IDE b) 1 c) 1	a) VGA b) - c) - d) 14"	a) - b) - c) - d) -	a) MS-DOS 5.0 b) da c) - d) -
PC-uri 386								
a) CSI Ltd. b) 90/12.10.45	a) TCR b) - c) 80386 d) 428.466 lei	a) - b) Intel c) 25 MHz d) -	a) 1 MB b) -	a) 120 MB (+ 169.265 lei) b) - c) 19 ms/ - d) -	a) IDE b) 1 c) 1	a) MDA b) - c) - d) 14"	a) - b) - c) - d) -	a) - b) - c) 18 luni d) -

a) Elinco S.A. b) 90/12.20.30	a) Sonic b) Ares USA c) 80386DX d) 940.000 lei	a) Mini Tower b) AMD/ Intel c) 40/ 33 MHz d) -	a) 4 MB b) 32 MB	a) 105 MB b) Quantum c) 14 ms/ 1:3 d) oricît	a) IDE/ SCSI b) 1 c) 1	a) SVGA 1024K b) 1024 x 768 c) 256 d) 14"	a) 1/2/0 b) ISA c) 0/6/1/0 d) 0/4/1/0	a) MS-DOS 5.0 b) da c) 18 luni d) 30 zile
c) MedComp DHS b) 90/15.56.53	a) Copam b) - c) 80386DX d) 4.475 DM	a) - b) Intel c) 8/33 MHz d) -	a) 4 MB b) 16 MB	a) 105 MB b) - c) 15 ms d) -	a) IDE b) 1 c) 1	a) VGA 512K b) 1024 x 768 c) 256 d) 14"	a) 1/2/0 b) AT-BUS c) - d) -	a) MS-DOS 5.0 MS-Windows 3.0 b) da c) - d) -
a) System Plus SRL b) 90/79.13.91	a) A-Tronic b) - c) 80386 d) 865.646 lei	a) - b) - c) 33 MHz d) -	a) 4 MB b) -	a) 120 MB b) - c) - d) -	a) IDE b) 1 c) 1	a) VGA 256K b) - c) - d) 14"	a) 1/2/0 b) - c) - d) -	a) - b) - c) - d) -
PC-uri 486SX								
a) Elinco S.A. b) 90/12.20.30	a) Ares b) USA c) 80486SX d) 1.150.000 lei	a) Full Tower b) Intel c) 20 MHz d) -	a) 4 MB b) 32 MB	a) 240 MB SCSI b) Quantum c) 13 ms/ 1:3 d) oricît	a) SCSI b) 1 c) 1)	a) SVGA 1024K b) 1024 x 768 c) 256 d) 14"	a) 1/2/0 b) ISA c) 0/6/2/0 d) 0/4/2/0	a) MS-DOS 5.0 b) da c) 18 luni d) 30 zile
a) MedComp DHS b) 90/15.56.53	a) Copam b) - c) 80486SX d) 3.910 DM	a) - b) Intel c) 8/20 MHz d) -	a) 4 MB b) 16 MB	a) 105 MB b) - c) 15 ms/ - d) -	a) IDE b) 1 c) 1	a) VGA 512K b) 1024 x 768 c) - d) 14"	a) 1/2/0 b) AT-BUS c) - d) -	a) MS-DOS 5.0 MS-Windows 3.0 b) da c) - d) -
PC-uri 486								
a) CSI Ltd. b) 90/12.10.45	a) TCR b) - c) 80486 + coprocesor Weitek 4167 d) 1.975.408 lei (inclusiv Streamer TEAC 60 MB)	a) Tower Case b) Intel c) 33 MHz d) -	a) 16 MB b) -	a) 80 MB, 128 KB cache b) - c) - d) -	a) - b) 1 c) 1	a) SVGA b) 1024 x 768 c) - d) 14"	a) - b) - c) - d) -	a) - b) - c) 18 luni d) -
a) Elinco S.A. b) 90/12.20.30	a)Ares b) USA c) 80486DX d) 1.390.000 lei	a) Full Tower b) Intel c) 50 MHz d) -	a) 4 MB b) 32 MB	a) 340 MB SCSI b) Quantum c) 11 ms/ 1:3 d) -	a) SCSI b) 1 c) 1	a) SVGA 1024K b) 1024 x 768 c) 256 d) 14"	a) 1/2/0 b) ISA c) 0/6/2/0 d) 0/4/2/0	a) MS-DOS 5.0 b) da c) 18 luni d) 30 zile
a) MedComp DHS b) 90/15.56.53	a) Copam b) - c) 80486DX d) 5.390 DM	a) - b) Intel c) 8/33 MHz d) -	a) 4 MB b) 16 MB	a) 105 MB b) - c) 15 ms/ - d) -	a) - b) 1 c) 1	a) VGA 512K b) 1024 x 768 c) - d) 14"	a) 1/2/0 b) - c) - d) -	a) MS-DOS 5.0 MS-Windows 3.0 b) da c) - d) -
a) System Plus SRL b) 90/79.13.91	a) A-Tronic b) - c) 80486 d) 1.132.383 lei	a) Full Tower b) Intel c) 33 MHz d) -	a) 4 MB, 64 KB cache b) -	a) 200 MB b) - c) - d) -	a) IDE b) 1 c) 1	a) VGA 256K b) - c) - d) 14"	a) 1/2/0 b) - c) - d) -	a) MS-DOS 5.0 b) da c) - d) -

Nota redacţiei:

Deoarece pînă la închiderea ediţiei am primit foarte puţine răspunsuri la ancheta iniţiată, vom reveni cu completările de rigoare de îndată ce vom dispune de informaţii suplimentare.

Rugăm firmele care doresc să ne ajute în prezentarea unei oglizi cît mai obiective a pieţei româneşti să ne comunice tipurile de echipamente de tehnică de calcul, pachete software, consumabile, cursuri, servicii, etc., care fac parte din oferta firmei proprii, pentru a le putea expedia următoarele chestionare.

Preţurile prezentate au un caracter pur informativ, fiind valabile în ziua în care noi am primit informaţiile. Pentru informaţii la zi vă rugăm să vă adresaţi direct firmelor în cauză.

PC-uri extensibile (2)

În prima parte a articolului, am prezentat conceptul de PC extensibil, PC conceput cu posibilitatea schimbării identităţii (tipul microprocesorului sau/şi frecvenţa tactului).

Ideea cea mai importantă în legătură cu PC-urile extensibile este aceea că, în majoritatea cazurilor, trăsătura de extensibilitate nu implică modificări esenţiale în planul preţului şi a performanţelor, ceea ce le face atrăgătoare şi pentru cei care nu au de gînd să-şi extindă PC-ul în viitor.

Important este de asemenea şi modul în care este implementată trăsătura de extensibilitate, deci tehnologia de extindere, care determină în mod hotărîtor restul aspectelor care caracterizează un PC extensibil: preţul configuraţiei de bază, preţul realizării extinderilor disponibile, ergonomia realizării unei extinderi.

Cea mai veche tehnologie de extindere este cea care derivă din utilizarea sertarelor cu fund de sertar pasiv. În această tehnologie, placa procesor este un modul ca oricare altul, putînd fi schimbată cu alta, cu condiţia asigurării compatibilităţii cu bus-ul. Sertarul cu fund pasiv a fost forma constructivă cea mai puţin utilizată. Creşterea gradului de integrare a condus la posibilitatea realizării unor produse monoplacă (single board computer) sau a înglobării principalelor funcţii ale unui sistem într-o placă de bază (placă sistem) care apare ca un fund de sertar activ pentru un sertar în care se introduc adaptoarele şi extensiile. Creşterea vitezei de lucru a componentelor a scos în evidenţă din ce în ce mai mult limitările de viteză impuse de fundul de sertar pasiv (8-12Mhz) ceea ce a dus la alegerea de noi soluţii constructive. Atît timp cît o soluţie monoplacă nu este, în general, realizabilă şi nici dorită (duce la scăderea flexibilităţii sistemului), soluţiile similare fundului de sertar sînt un rău necesar. Esenţial este ca sistemul să fie în aşa fel croit încît la nivelul interconexiunilor dintre module, comunicaţia sa să se poată face şi la o viteză mai mică fără a fi afectate performanţele sistemului.

În cazul PC-urilor, bus-ul sistem standard (AT BUS) care era satisfăcător ca şi viteză în momentul apariţiei PC AT-urilor, a devenit între timp prea "îngust". Ca atare, mişcarea IBM-ului cu bus-ul MCA, poate părea prea bruscă, dar este justificată.

Revenind la tehnologia sertarului cu fund de sertar pasiv, trebuie spus că aceasta a fost promiţătoare în cazul maşinilor CP/M de 8 biţi, dar a căzut în dizgraţie o dată cu apariţia PC-urilor, fiind cunoscute foarte puţine PC-uri în care să fie utilizată: Wang Professional PC în 1985, Tyse şi Zenith în 1987 şi, de ce nu, TELEROM PC fabricat cîndva la IEIA Cluj.

O altă tehnologie de extindere, utilizată efectiv, se bazează pe ideea "spargerii" plăcii de bază în două părţi: una cu componentele independente de tipul şi viteza microprocesorului şi alta cu componentele specifice printre care microprocesorul însuşi. În realitate, nu se poate face o separare atît de strictă între ceea ce este dependent şi ceea ce este independent de tipul şi viteza microprocesorului, trebuind să fie făcute compromisuri legate de realizarea unui raport preţ/performanţă dorit.

Prima parte este noua placă de bază. A doua este modulul de extensie sau placa procesor. Modulul de extensie se introduce într-un slot special, ale cărui caracteristici depind de felul în care s-a făcut "spargerea". Principala problemă o constituie relaţia dintre procesor şi memorie. Soluţia punerii memoriei pe modulul de extensie, deci implicit, cînd este cazul, şi a cache-ului, este mai scumpă dar, în cazul unor fabricanţi, se poate obţine o reducere de preţ prin recuperarea de către aceştia a unor componente.

Dacă memoria rămîne pe placa de bază, atunci conexiunea trebuie să fie de mare viteză şi cu o lăţime a bus-ului la nivelul microprocesorului celui mai puternic. Dacă nu se respectă aceste condiţii, performanţele sistemului vor fi afectate drastic. Existenţa unui cache pe placa procesor (la 486 este chiar pe chip), atenuează scăderea de performanţă datorată unei conexiuni procesor-memorie necorespunzătoare.

În cazul extinderii de la 386 la 486, unii producători au ales soluţia înlocuirii microprocesorului 386 cu un modul care conţine microprocesorul 486 şi cîteva circuite de adaptare care fac ca modulul să fie compatibil la nivelul pinilor cu un 386. Acest modul se introduce direct în soclul microprocesorului 386.

Schimbarea microprocesorului poate conduce la necesitatea unor schimbări în BIOS. Problema este rezolvată în mod diferit de la un fabricant la altul. Soluţiile cele mai comode pentru utilizator sînt cele care nu-i cer acestuia nici o operaţie suplimentară în afara schimbării modulului procesor. Vă prezentăm două dintre acestea: ROM-urile de BIOS sînt pe modulul de extensie sau BIOS-ul configuraţiei iniţiale se poate adapta tuturor procesoarelor din extensiile posibile. O soluţie interesantă este cea cu EPROM care cere utilizatorului să ruleze un program de modificare a conţinutului EPROM-ului funcţie de tipul microprocesorului instalat. În fine, se utilizează şi soluţia banală a schimbării efective a ROM-urilor pe placa de bază.

În continuare vom prezenta cîteva tipuri de PC-uri extensibile. Ne vom referi în special la caracteristicile legate de extensibilitate. În general, PC-urile prezentate sînt fabricate de firme cu un bun renume pe piaţa occidentală.

1. ALR Business VEISA

Este produs de firma ALR (Advanced Logic Research). În configuraţia de bază conţine o placă procesor 386/33. Sînt disponibile două module de extensie cu 486/25, respectiv 486/33. ALR practică o reducere de preţ pentru cei care trimit firmei vechiul modul. Pentru extensii, PC-ul este prevăzut cu trei sloturi speciale. Unul este ocupat de placa procesor. În celelalte două se pot introduce opţional un modul de cache şi un modul de extensie de memorie.

2. ALR Modular Processor System

Acest tip de PC are opţiuni în toate trei arhitecturile: ISA, EISA şi MCA. Sînt disponibile module procesor pentru 386/25, 386/33, 486/25 şi 486/33. De curînd a apărut şi un modul cu 486SX. Pe placa procesor exista procesul, soclu pentru coprocesor şi, la 386/33, so-

clu pentru chip-ul Intel Smartcache. Şi în cazul acestor PC-uri, ALR practică reduceri de preţ în cazul recuperării procesorului vechi.

3. ALR SX Power Flex

Acest tip de PC extensibil, produs de firma ALR, oferă extensibilitate, prin intermediul modulelor procesor, de la 286 la 486SX trecînd prin 386SX/16 şi 386SX/20. Datorită faptului că memoria este pe placa de bază, iar legătura cu modulul procesor se face printr-un bus de 16 biţi (configuraţia cu 486DX/25 are performanţe foarte slabe în raport cu PC-urile 486DX/25 adevărate, fiind comparabile cu cele ale unui 386DX/33. Cu excepţia microprocesorului 286, restul sînt recuperate contra cost de ALR în momentul cumpărării unui modul de extensie.

Toate PC-urile extensibile produse de ALR se caracterizează prin modul simplu de realizare a extensiei. Singurele operaţii necesare sînt scoaterea modulului vechi şi introducerea celui nou.

4. Amkly 386/25e

Produs de firma Amkly Systems, 386/25e este un PC extensibil cu bus EISA. Are două opţiuni: 386DX/25 şi 486DX/25. Extensia se realizează prin plăci procesor care conţin pe lîngă microprocesor, patru slot-uri pentru SIMM-uri şi controller-ul de cache Intel 82359. Modulul 386 conţine şi un soclu pentru coprocesor, preţul extinderii poate fi redus prin trimiterea la firmă a vechiului modul 386DX/25 de pe care au fost mutate SIMM-urile pe noul modul, 386/25e este proiectat în aşa fel încît va putea găzdui, în viitor, şi extensii mai rapide de 33, 50 sau 66 Mhz.

5. AST Premium II

Produse de firma AST Research, aceste PC-uri extensibile oferă performanţe peste medie (cu excepţia adaptorului video) pentru următoarele module procesor: 386SX/20, 386DX/33, 486SX/20 şi 486DX/33. Modulele procesor, numite FAST board, se introduc în oricare din cele trei slot-uri ISA extinse la 32 biţi (arhitectura CUPID, proprietatea firmei AST). Pe placa procesor se pot pune pînă la 16 Mocteţi de RAM. În plus se pot pune plăci de memorie în cele două slot-uri CUPID libere. Extinderea constă în simpla schimbare a plăcii procesor. BIOS-ul detectează tipul FASTboard-ului instalat, luînd măsurile care decurg din modificarea tipului/vitezei microprocesorului.

6. Copam B Serie

Firma Copam oferă una dintre cele mai largi game de extensii: 386SX/16, 386DX/25, 386DX/33, 486DX/25 şi promite 386SX/20, 386DX/40 şi 486DX/50. Tehnologia de extindere este cea a modulelor procesor care se introduc într-un slot special. Pe modulul de extensie se află, în afară de microprocesor, un soclu pentru coprocesor, memoria RAM, BIOS-ul şi CMOS-ul, practic toate circuitele a căror funcţionalitate trebuie să ţină pasul cu microprocesorul. Este remarcabil faptul că preţul modulelor de extensie nu este cu mult mai mic decît cel al firmelor care pun pe modulul de extensie numai procesorul şi soclul coprocesorului.

7. Dell System 333P

Produs de firma Dell Computer Corp., 333P conţine, în configuraţia de bază, un 386DX/33 şi realizează performanţe modeste. Cu un modul suplimentar de cache, performanţele cresc spectaculos, ducînd sistemul între maşinile bune. Microprocesorul 386DX este înglobat în placa de bază. Pentru extindere la 486DX/33, trebuie achiziţionată o placă procesor care se introduce într-un slot al plăcii de bază. Odată instalat, microprocesorul 486 va înhiba microprocesorul 386 de pe placa de bază, nefiind practic nevoie ca acesta să fie scos. Dacă sistemul iniţial conţine cache, acesta devine inutil ba chiar trebuie scos pentru a face loc modulului procesor 486. Proiectanţii au mizat pe cache-ul intern din 486. Trebuie scos de asemenea şi coprocesorul 387 dacă acesta există. În plus, trebuie schimbat şi BIOS-ul. Se observă că, la acest tip de PC-uri, realizarea extinderii este ceva mai laborioasă şi presupune scoaterea din sistem a unor componente pe care Dell nu le recuperează contra cost.

8. Everex STEP386 cu InSTEP486

Firma Everex Systems produce PC-ul STEP386 în versiunile 20, 25 şi 33 Mhz. Aceste PC-uri pot fi extinse la 486 de frecvenţă corespunzătoare. Extinderea se face prin introducerea plăcii InSTEP486 în soclul microprocesorului 386. Această operaţie este foarte dificilă şi cu un mare risc de distrugere a modulului de extensie care costă "numai" 2.999$. În plus, trebuie schimbat şi BIOS-ul. Cine poate trece cu vederea aceste dificultăţi, va beneficia de un 386 şi apoi de un 486 de foarte bună calitate.

9. Tandon Option

Tandon oferă o serie completă de module procesor: 286/12, 386SX/20 fără şi cu cahe, 486SX/20 şi 486DX/33. PC-urile extensibile Tandon se remarcă prin procedura extrem de simplă de schimbare a plăcii procesor. Modulul procesor este închis într-o cutie metalică, schimbarea fiind la fel de simplă ca şi schimbarea unei dischete în drive. Deoarece memoria este situată pe placa de bază şi bus-ul de date este de 16 biţi, configuraţia 486/33 va avea performanţe mai scăzute în lucrul cu memoria.

10. Wearnes-seria Boldline

PC-urile Wearnes sînt realizate pe baza unor proiecte ALR. Seria Boldline consta din PC-uri extensibile şi neextensibile. În configuraţia de bază, PC-urile extensibile vin cu 286/12.5, 286/16, 386SX/16, 386SX/20, 386SX/25 sau 486SX/20. Placa de bază este prevăzută cu un slot dedicat de 104 contacte în care se pot introduce modulele procesor cu 386SX/16, 386SX/20, 386SX/25, 486SX/20 şi 486SX/25. Evident că pentru o anumită configuraţie de bază au sens numai extensiile mai puternice.

Firma DARIAN ROM SUISSE SRL va comercializa în viitor calculatoare Wearnes.

Mitică Caraiani,
DARIAN ROM SUISSE SRL,
tel. 95/123611, fax. 95/124567

Sisteme multimedia. Metode de acces la text. Metoda semnăturii.

În urma acordului de colaborare ştiinţifică şi culturală semnat în anul 1990 de Germania şi România, fundaţia *Deutscher Akademischer Austauschdienst (DAAD)* cu sediul la Bonn, a acordat României burse de specializare în Germania în anii 1990-1992. Pentru a fi acceptată candidatura, era necesară o invitaţie din partea unui institut sau a unei universităţi cu indicarea temei şi a utilităţii la întoarcerea în ţară. În urma unei corespondenţe cu diferite universităţi vest-germane, am primit din partea *Prof. Dr. H.J. Appelrath* invitaţia de a participa timp de două luni la proiectele colectivului Sisteme informatice al *Facultăţii de Informatică a Universităţii Oldenburg.*

Tema de cercetare la care am participat poartă numele de **Multimedia Archiving and Retrieval System (MARS)** şi este parte a proiectului **A Multimedia Filing System (MULTOS)**, proiect **ESPRIT Nr. 28**. În ciudaa timpului scurt, programul de instruire la care am participat a fost deosebit de consistent.

Universitatea Oldenburg este o universitate de capacitate medie avînd facultăţi cu specificul Ştiinţele Naturii: pictură, muzică, pedagogie, sociologie, filologie (literatură germană şi engleză), istorie, geografie, biologie, informatică, religie etc.

Dotările pentru studenţi sînt deosebite: piscină, terenuri de sport acoperite, cantină, bufet, librărie şi o bibliotecă extraordinară desfăşurată pe mai multe etaje cu posibilitatea computerizată de consultare, camere de proiecţie şi colecţii de reviste şi rapoarte la consfătuiri ştiinţifice.

În cadrul Facultăţii de Informatică sînt mai multe colective de cercetare în domeniile informatică teoretică şi aplicativă.

Proiectul **MARS** urmăreşte construirea de modele pentru diferite structuri semantice ale documentelor multimedia, realizînd administrarea automată a datelor.

Conţinutul unui document multimedia depinde de aplicaţie din mai multe puncte de vedere:

- referitor la prezentarea sa în pagină: text, imagine etc. (de exemplu standarde Post Script sau ODA);
- referitor la structura sintactică: forma paragrafelor, capitolelor, titlurilor etc. (de exemplu standardele de structuri logice SGML sau ODA);
- referitor la interpretarea semnatică a sa.

Documentele multimedia în aplicaţiile moderne de birotică necesită o modelare flexibilă care să permită accesul rapid la informaţie, abstractizarea conţinutului într-o formă cît mai apropiată de limbajul natural. O scrisoare spre exemplu are de obicei o prezentare standard în pagină: dimensiunea paginii, tipul font-urilor, o structură sintactică standard: antet, persoana căreia îi este adresată, formula de adresare, conţinut, semnătură şi o structură semantică standard în cazul în care este o ofertă, un refuz, o felicitare etc.

Proiectul **MARS** urmăreşte tipizarea reprezentării documentelor pe diferite clase (documente de afaceri, oferte, scrisori) şi în cadrul fiecării clase pe diferite scheme, toate aceste structuri aflîndu-se într-un catalog din care pot fi extrase modelele dorite. Pentru crearea unui nou document (o scrisoare de răspuns de exemplu) este ales un model corespunzător completat cu "valori" extrase din alte documente (pentru exemplul de mai sus: scrisoarea cerere, baza de date cu informaţiile necesare etc.). Bineînţeles acest program este în strînsă legătură cu sisteme de baze de date pe care le interoghează pentru a extrage informaţii şi pe care le completează cu noi date. Modul de regăsire este clasic, cu ajutorul unui limbaj de interogare în care se adresează cererea. Pentru regăsirea rapidă a datelor este posibilă indicarea valorilor ataşate conceptelor. Prin gruparea şi generalizarea structurilor se poate organiza o structură ierarhică ce permite regăsirea eficientă.

Problema ce se pune este cea a optimizării lucrului: timp de răspuns scurt şi în special posibilităţi de arhivare. În multe situaţii nu este suficientă o bază de date la care regăsirea să se facă după dicţionar sau cuvînte cheie. În cazul documentelor multimedia ce sînt integral arhivate problema regăsirilor se complică datorită dimensiunii lor mari şi a faptului că trebuie construite strategii diferite pentru text, imagine, sunet.

Programul MARS se găseşte la Universitatea Oldenburg în faza de lucru. Programarea se face în MODULA2 şi C++ pe staţii de lucru SUN sub sistemul de operare UNIX. Pentru eficienţa implementării sub MODULA2, în cadrul colectivului condus de prof. dr. H.J. Appelrath, a fost realizată o bibliotecă de module pentru MODULA2 numită OM2Lib (Oldenburg Modula2 Library), portabilă pe diferite echipamente. Biblioteca OM2Lib conţine modulul matematic, module pentru operaţii de I/O, lucrul cu terminalele standard sau nestandard, erori, lucrul cu memoria (alocare, dezalocare), cu şiruri, liste, mulţimi, fişiere, etc. Pentru că este de curînd dată în exploatare (9.10.1991) şi deci încă în faza de testare, am ajutat la testare descoperind cu ocazia implementării erori în modulele OM2Lib.

Prof. dr. H.J. Applerath împreună cu responsabilul proiectului MARS, dr. informatician H. Eirund, mi-au propus spre rezolvare problema regăsirilor de text prin metoda semnăturii (Signature Methode).

În ultima decadă au apărut numeroase situaţii (biblioteci computerizate foarte vaste, sisteme informatice de birou conţinînd documente multimedia) care pun serioase probleme regăsirii manuale a textului.

Fox Pro 2.0

DARIAN ROM SUISSE SRL
3400 CLUJ-NAPOCA
str. Observatorului nr. 146/15
tel. : 95-123611
fax: 95-124567

Pe o bază de date de 200MB, avînd milioane de articole, FOXPRO pe PC este mai rapid de 4 ori decît DB2/IBM pe un mainframe, iar diferenţa faţă de alte sisteme de gestiune ca Paradox sau DBase este enormă. Toate acestea datorită noii tehnologii de acces la date RUSHMORE.

Utilitarele sale de generaţia a patra, orientate pe obiecte, (generator de ecrane, de rapoarte, de meniuri şi proiecte) cresc extraordinar viteza de dezvoltare a aplicaţiilor. Şi toate acestea generează programe sursă FOX, accesibile utilizatorilor !

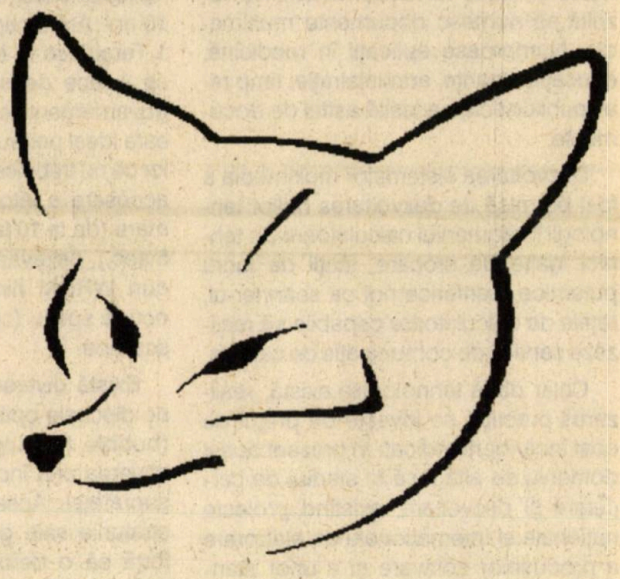
Ce este nou în FOXPRO 2.0 ?

Noul sistem de gestiune a memoriei asigură o exploatare optimă, astfel încît viteza creşte atît la sisteme mono- cît şi multiuser. Volumul indexării compuse reprezintă un sfert din cel al sistemelor similare şi totuşi este mult mai rapid.

RQBE / Relational Query-By-Example / asigură o interfaţă simplă cu ajutorul căreia, doar cu cîteva mişcări de mouse, obineţi informaţiile de care aveţi nevoie. FOXPRO va genera intrucţiunea SELECT/SQL corespunzătoare, care împreună cu celelalte comenzi SQL: INSERT şi CREATE TABLE pot fi folosite în aplicaţiile FOX.

FOXPRO 2.0 a fost dotat cu peste o sută de comenzi şi funcţii noi, ca de exemplu relaţia "one-to-many-to-many", noi metode de sortare, sau comanda BROWSE FOR care ar putea să fie ea singură baza unei noi aplicaţii.

Dezvoltarea proiectelor se realizează simplu şi eficient cu noul "project-manager", care urmăreşte fişierele folosite, înglobează în sistem toate modificările sau reconstruieşte întreaga aplicaţie. Generarea programelor EXE şi distribuirea produselor realizate este posibilă prin modulul opţional Distribution Kit.



Este posibil importul datelor dintr-o multitudine de formate : PARADOX, MULTIPLAN, LOTUS, SYMPHONY, EXCEL etc.

Prin produsele opţionale ale familiei FOX : Fox Graph, Distribution Kit sau Application Program Interface, Fox Pro 2.0 devine un instrument eficient şi complet pentru dezvoltarea aplicaţiilor.

Trimiteţi-mi dischetele demonstrative FOXPRO 2.0

NUME . . . . . .

FUNCŢIA . . . . . .

FIRMA . . . . . .

ADRESA . . . . . .

. . . . . .

TELEFON . . . . . .

FAX . . . . . .

Format dischetă 3,5" |_| 5,25" |_|

Dischetele demonstrative pot fi obţinute gratuit numai în schimbul acestui talon!

Înainte de a descrie lucrarea, socotesc că este foarte util să prezint cîteva cuvinte despre suportul teoretic al acesteia.

Sistemele multimedia se definesc ca sisteme capabile să combine şi să alăture o mare varietate de tipuri de informaţie: text, grafică, voce, imagine, audio şi video iar documentele ce rezultă se numesc documente multimedia. Numeroase aplicaţii în medicină, educaţie, finanţe, administraţie, timp real, publicistică, necesită astfel de documente.

Dezvoltarea sistemelor multimedia a fost permisă de dezvoltarea noilor tehnologii în domeniul calculatoarelor: tehnici înalte de stocare, staţii de lucru puternice, periferice noi ca scanner-ul, reţele de calculatoare capabile să realizeze servicii de comunicaţie de calitate.

Chiar dacă tehnologiile există, realizarea practică se loveşte de preţul de cost încă foarte ridicat. În prezent acest domeniu se află încă în stadiul de cercetare şi dezvoltare, existînd proiecte naţionale şi internaţionale de elaborare a produselor software şi a unor standardelor internaţionale pentru tehnologii şi comunicaţii.

O primă problemă care s-a pus a fost cea legată de existenţa unei tehnologii de stocare capabilă să reţină un volum mare de date. Începînd cu anul 1980, discurile optice au reprezentat o soluţie a acestei probleme, ţinînd cont de faptul că pe un disc optic cu diametrul de 12" pot fi stocaţi cîţiva gigabytes.

Se realizează discuri optice în trei variante: discuri optice read only (Compact Disc Read Only Memory CD-ROM), discuri optice Write Once, Read Many Times (WORM) şi discuri optice mixte write/read.

CD-ROM reprezintă o memorie externă de mare capacitate şi preţ scăzut (10 $ la nivelul anului 1987). Timpul de acces este de 200 msec putînd stoca de la unu pînă la cîţiva gigabytes. Este folosit pentru stocarea de baze de date care nu necesită actualizări. Pentru prima dată a fost utilizat în publicistică. Mecansimul optic de citire este la distanţă de disc, care este îmbrăcat într-o folie de plastic sau sticlă, astfel încît informaţia este protejată la deteriorare fizică. ISO (International Standards Organisation) a adoptat un standard pentru această tehnologie.

WORM reprezintă un mediu sigur pentru arhivare. Permite scrierea doar o singură dată avînd acces cvasi direct şi are un timp de întrebuinţare foarte lung. Nu permite actualizări, ştergeri, deci discul nu poate fi reutilizat. Multe firme garantează acest disc pentru mai mult de 10 ani. Are o capacitate de mai mult de 1 Terabytes (1 Milion de Mbytes) ceea ce îl face deosebit de util în aplicaţii guvernamentale sau de afaceri. WROM este ideal pentru memorarea informaţiilor ce nu trebuiesc actualizate. Viteza de accesare a informaţiei, nu este foarte mare (de la 10 la 13 biţi în mai puţin de 5 sec.), dezavantajul unor tipuri de discuri WROM fiind legat de faptul că ocupă spaţiu (conţin 12 discuri) şi sînt scumpe.

Există diverse tehnologii de scriere pe discurile optice. Una dintre acestea (bubble forming process) realizează scrierea prin încălzirea cu raza laser a suprafeţei. Aceasta formează în substraturile sale gaz care are suficientă forţă să o deformeze. Raza de citire detectează deformările suprafeţei. O altă tehnică topeşte suprafaţa de telurium cu raza laser, citirea făcîndu-se prin detectarea schimbării reflexivităţii suprafeţei.

Discurile optice reutilizabile (rewritable) sînt cele mai recent apărute. În mod obişnuit au o capacitate de 650 Mbytes, unele tipuri ajungînd pînă la 30-50 Gbytes. Au o dimensiune de 5,25 inci. Tehnologia de realizare este fie prin impurificarea teluriumului astfel încît ştergerea se realizează prin crearea de curenţi în suprafaţa de citire-scriere ce o nivelează, fie prin tehnici combinate optice şi magnetice. ISO şi ANSI (American National Standard Information) lucrează la standarduri pentru WROM.

Un prim mare avantaj al discurilor optice este capacitatea mare de stocare. Al doilea avantaj este legat de faptul că se pot deteriora mai greu decît cele magnetice. Domeniul de cercetare în această privinţă este legat de realizarea de discuri optice reutilizabile care să permită rescrierea directă fără ştergere prealabilă. Apariţia tehnologiilor pentru discuri optice a permis dezvoltarea sistemelor informatice multimedia.

Pentru o mare varietate de aplicaţii, utilizarea principală a calculatorului este în domeniul stocării şi manipulării unui mare volum de informaţii. Calculatoarele sînt dotate cu sisteme ce permit utilizatorilor să stocheze, regăsească şi să şteargă informaţia. În mod obişnuit sistemele de operare sub care se lucrează permit stocarea datelor într-o structură ierarhică formată din cataloage (directory) şi fişiere. Într-un fişier se poate afla informaţia reprezentată de diferite medii (text, imagine, sunet). Clasic, regăsirea se face la nivel de fişier prin parcurgerea liniară a cataloagelor. În ultimii ani, noile tehnologii de stocare şi staţiile de lucru puternice au permis extinderea acestei regăsiri liniare. În anii 1960 a apărut noţiunea de **hypertext** definită de *Ted Nelson* ca "o combinaţie între text în limbaj natural şi capacităţile calculatorului pentru lucrul interactiv, display dinamic a unui text neliniar care nu poate fi tipărit convenţional pe o pagină convenţională".

Hypertext poate fi definit (*Jef Conklin*) ca o reprezentare schematică formată dintr-o reţea de obiecte dintr-o bază de date şi o metodă de acces la baza de date. Pe de altă parte, metodele de acces presupun un sistem de ferestre şi imagini pe ecran asociate obiectelor din baza de date.

În momentul în care au apărut calculatoare capabile de a stoca şi alte tipuri de informaţie, problema regăsirilor a fost pusă pentru diferite medii, apărînd şi dezvoltîndu-se sistemele hypermedia. Calculatoarele bazate pe sisteme hypermedia sînt definite ca o extensie a hypertextului ce încorporează alte medii.

Sistemele hypermedia permit utilizatorilor crearea de informaţie ce să includă text, voce, imagine, grafică, video, sunet.

În momentul actual există mai multe centre în lume se utilizează astfel de sisteme: administraţie, în biblioteci, sisteme de instruire etc.

Tehnologiile pentru sistemele de gestiune a bazelor de date curente permit modelarea datelor şi transferul către zona CAD-CAM sau aplicaţii medicale. Aceste tipuri de aplicaţii necesită lucrul cu mai multe tipuri de date. De aceea au fost proiectate sisteme de gestiune a

bazelor de date multimedia (**MDBMS - Multimedia Data Basis Management Systems**). MDBMS permite nu numai stocarea de date de diverse tipuri dar şi operaţii sofisticate de regăsire.

În proiectarea MDBMS s-a pornit de la baze de date relaţionale care au fost reorganizate pentru a permite stocarea diferitelor tipuri de date. Altă modalitate de proiectare este construirea de MDBMS pornind de la baze de date orientate pe obiect (Object Oriented Databasis). Un model unic nu a fost încă creat, acest domeniu aflîndu-se în faza de cercetare.

Privind implementările actuale, se observă că problema ce se ridică este distribuţia datelor multimedia.

Indiferent de tipul datelor, caracteristic MDBMS sînt dimensiunile foarte mari ale datelor. Pentru regăsiri se folosesc strategii adecvate mediului, diferite pentru text, imagine sau sunet, toate urmărind însă o regăsire rapidă şi eficientă. Cerinţa formulată pentu regăsire conţine valori ale diverselor atribute (cuvinte cheie, informaţii despre pacient, numele cîmpurilor etc.) şi combină operatori Booleeni AND, OR, NOT.

Multe dintre metodele de acces la informaţie au fost deja propuse în literatura sistemelor de gestiune a bazelor de date. În general, **metodele de regăsire a textului** se clasifică în patru categorii: parcurgerea (scaning) întregului text, inversiunea, metode bazate pe multiatribut şi metode bazate pe clustering.

Metodele de parcurgere integrală a textului sînt uşor de implementat dar bineînţeles, sînt cele mai lente. Problema care se pune este regăsirea unui şir de caractere (string).

Dacă întrebarea este o expresie Booleană complicată, trebuie introdus un nou pas în algoritmul de rezolvare, pas care să determine care dintre termenii găsiţi satisfac expresia Booleană. Avantajele metodelor de parcurgere completă a textului este că nu necesită spaţiu de memorie şi nu pun probleme în cazul inserărilor sau actualizărilor.

Datorită timpului de răspuns mare, aceste metode nu sînt recomandate în cazul bazelor de date mari.

Inversiunea termenilor presupune reprezentarea fiecărui document ca o listă de cuvinte (chei) care descriu conţinutul documentului. Aceste chei sînt stocate într-un fişier ataşat documentului numit fişier invers, cheile fiind stocate alfabetic. Dezavantajele sînt spaţiul mare de stocare (50-300% din fişierul original) şi problemele legate de actualizări sau reorganizări. Avantajele sînt implementarea facilă şi rapiditatea regăsirilor. Din aceste motive, multe sisteme de gestiune a bazelor de date au adoptat această metodă (DIALOG, MEDLARS, ORBIT, STAIRS, CDSISIS).

Metodele de regăsire cu multiatribut se bazează pe metoda semnăturii. Să presupunem că avem o înregistrare formată din trei atribute (chei): "parcurgerea", "integrală", "textului". Fiecare atribut este codificat cu ajutorul unei funcţii într-un şir de F bits, dintre care m bits sînt 1, rezultînd "semnătura" cuvintelor.

Aceste semnături sînt stocate într-un fişier separat numit "fişierul semnătură". Nu există nici o restricţie în legătură cu dimensiunea documentului ce trebuie arhivat. Dezavantajul acestor metode este timpul de răspuns lent în cazul fişierelor mari, de aceea trebuie alese variante optime pentru implementare. Avantajul este eficienţa în cazul actualizărilor, abilitatea de a formula întrebări doar cu părţi din cuvînt (trunchieri ale atributelor), abilitatea de a suporta creşterea dimensiunii fişierelor, precizie la regăsire.

Metodele bazate pe clustering au la bază ideea grupării documentelor similare pentru a forma clusters (ciorchini). Ipoteza de la care se porneşte este aceea că documentele ce au un conţinut sau o structură similară răspund în general aceloraşi cereri. Gruparea poate fi făcută şi termenilor din cadrul unui document. Fiecare document este reprezentat ca un vector tridimensional şi anumite cuvinte cheie îi sînt asociate cu ajutorul unei funcţii de indexare. Numărul cuvintelor cheie dă dimensiunea vectorului.

Accesul la document se face prin indirectarea într-un spaţiu t-dimensional. Următorul pas este gruparea vectorilor în clusters, indirectarea făcîndu-se în aceste grupe. Regăsirile sînt de obicei accelerate prin gruparea în clusters. Dezavantajul este legat de faptul că nu există un algoritm matematic pentru determinarea numărului de clusters (determinarea este făcută după o metodă empirică).

Recent au fost adoptate **soluţii hardware** pentru regăsirea textului. Acestea sînt mult mai atractive, în special în condiţiile descreşterii preţului de cost la echipamentele hardware. În momentul de faţă, nu există pentru aceste metode performanţe legate de timpul de răspuns.

Tema de cercetare propusă, **Regăsirea textului prin metoda semnăturii,** urmăreşte rezolvarea problemei regăsirilor de text prin metoda cu multiatribut a semnăturii şi a fost dezvoltată în MODULA2 pe sistem UNIX, utilizînd modulele bibliotecii MODULA2, OM2Lib. Programul a fost realizat ca un tip de date abstracte în trei module: Signature, Basis şi Hash.

Modulul Signature este de fapt tipul de date abstracte pentru acces şi extracţie de cuvinte dintr-un fişier semnătură, celelalte două fiind module ajutătoare de lucru: pentru constante şi calcule.

Utilizînd modulul Signature, este posibilă crearea de fişiere semnătură cu un număr nelimitat de semnături, inserarea documentelor în fişiere semnătură (arhive), actualizări şi bineînţeles regăsiri de text.

S-a pornit de la ideea de a avea arhivat întregul text semnificativ al unui document, excluzînd de la arhivare prepoziţii, pronume, articole etc.

Există mai multe metode de obţinere a semnăturii cunoscute în literatura de specialitate prin iniţialele cuvintelor ce le descriu:

- WS - Word signature,
- SC - superimposed coding,
- BC - bit block compression,
- RL - run length encoding,
- EN - entropy-based bounds,
- VBC - variabile bit-block - compresion.

Descrierea acestor metode necesită o dizertaţie teoretică consistentă, chiar o lucrare separată cu acest subiect. Fără a arăta cum s-a ajuns la această concluzie, trebuie spus că cea mai eficientă metodă din punctul de vedere al timpului de răspuns este SC, iar cea mai economică din punctul de vedere al spaţiului utilizat RL. BC şi VBC realizează un compromis între SC şi RL.

Din aceste metode, am ales pentru implementare metoda SC.

Proiectarea şi programarea tipului de date abstracte Signature a ridicat mai multe probleme interesante: alegerea algoritmului, programarea cît mai independentă de maşină, soluţii optime de implementare.

La alegerea algoritmului am fost ajutată de existenţa bibliotecii universităţii unde am găsit material documentar foarte interesant. Pentru obţinerea semnăturii am aplicat o metodă după modelul propus de *Christodoulakis* şi *Faloutsos* (1984) bazat pe ideea suprapunerii codului a lui *Mooers* (1949). Problemei timpului de acces şi a reducerii spaţiului de arhivare i-am găsit ca soluţie utilizarea unui fişier listă cu numele documentelor, ataşat fişierului semnătură. În cazul inserării sau al eliminării trebui întîi verificat dacă respectivul document se găseşte sau nu arhivat în fişierul semnătură şi, pentru că parcurgerea acestuia din urmă necesită prea mult timp, verificarea se face în fişierul listă. De asemenea, fiecărei semnături îi este ataşat un cod ce reprezintă documentul căreia îi aparţine şi legătura cod - nume document se face în fişierul listă. Altfel, ataşarea la semnătură chiar a numelui fişierului ar însemna creşterea substanţială a dimensiunilor arhivei.

Asupra fiecărui cuvînt al documentului, cu excepţia cuvintelor excluse de la arhivare, se aplică o hash-funcţie ce dă semnătura cuvîntului. Metoda SC presupune obţinerea de semnături nu pe cuvînt ci pe blocuri de text prin suprapunerea cu o funcţie Booleană OR a semnăturilor fiecărui cuvînt, obţinîndu-se pentru arhivare o semnătură pentru întregul bloc. În felul acesta se rezolvă atît problema spaţiului cît şi a timpului de acces. Lucrarea urmează să fie continuată pentru metoda BC.

Procedurile conţinute în tipul de date abstracte Signature sînt:

Init - crează un fişier cu semnături conţinînd în prima înregistrare metoda de lucru (SC sau BC), numărul F de bits din tiparul semnăturii, numărul m de 1 bits şi numărul D de cuvinte a căror semnătură se suprapune pentru a forma semnătura blocului. Prin procedura Init este creat şi un fişier cu numele nume-fişier-signature.lst care conţine numele documentelor arhivate în fişierul semnătură, legătura între cele două făcîndu-se printr-un cîmp de cod ataşat fiecărei semnături a documentului cît şi numelui din listă.

Insert - inserează un document în fişierul semnătură şi numele lui în fişierul listă.

Remove - elimină din fişierul semnătură un fişier document.

open - deschide atît fişierul semnătură cît şi fişierul listă de documente.

Close - închide fişierele deschise.

Get Candidate Simple - este procedura care la formularea unei cerinţe (un cuvînt) dă o listă cu documentele conţinînd această cerinţă.

Write Result Proc - este o procedură utilizată de Get Candidate Simple pentru afişarea listei de fişiere document.

Rezolvînd această temă, am avut de cîştigat foarte mult atît în domeniul teoretic cît şi în cel al proiectării şi programării.

Ceea ce este prezentat în acest raport este doar un rezumat al activităţii desfăşurate la Oldenburg, cît şi o mică parte a suportului teoretic care stă la baza metodelor moderne de regăsire a textului.

Timpul fiind foarte scurt (doar 2 luni dintre care primele două săptămîni au însemnat adaptarea la lucrul cu MODULA2 şi sub sistem UNIX), ceea ce am realizat este doar o parte a temei de cercetare. Aceasta se poate dezvolta cu adăugarea unei proceduri complete pentru cerinţe cu operatori Booleni AND, OR şi NOT, a unei proceduri de eliminare a paşilor falşi şi completarea procedurilor existente pentru alte metode (BC şi RL).

Dată fiind lipsa de documentaţie în acest domeniu, prezentarea detaliată a metodelor de acces pentru text ar fi de un mare interes.

De asemenea, seminarii sau informări cu tema "Documente multimedia" ar fi utile ţinînd seama că viitorul sistemelor informatice de birou, al bibliotecilor şi al instituţilor de informare-documentare este legat de documentele multimedia, fiind trecută în prezent atît bariera tehnologică (există acum discuri optice de capacitate mare şi timp de acces rapid, sisteme de comunicaţie etc.) cît şi cea a lipsei aplicaţiilor.

11.01.1992
Dipl. Ing. Mirela Guţică

Bibliografie:

» 1. Archivierung multimediales Dokumente H. J. Appelrath, H. Eirund - 1990
» 2. Acces Methodes for Text - Christos Faloutsos - 1985
» 3. Multimedia Services and Applications Luis Orozco Barbosa, Nicolas D. Georganas - Telecomunications Research Institute of Ontario Departament of Electrical Engineering.
» 4. Design of Office Informations Systems C. A. Ellis, N. Naffah - Springer-Verlag Berlin Heidelberg New York - 1987
» 5. Signature - Abstract Data Type for Archiving and Acces of the Text - Mirela Guţică - Oldenburg 1991
» 6. Bericht über das Thema "Textsignaturen" - Mirela Guţică - Oldenburg 1991

*** Gratis * Gratis * Gratis ***

SokySoft vă oferă gratuit pachetul de rutine

DERIVATA 1.0

pentru derivarea simbolică a funcţiilor. Cei care doresc să intre în posesia acestor rutine nu trebuie să achite decît contravaloarea unei dischete şi a cheltuielilor de expediţie, sau să trimită o dischetă proprie, goală, pe adresa:

Socaciu Ioan Tiberiu,
str. Arieşului, nr. 102,
bl. B2, sc. II, ap. 55,
3400 Cluj - Napoca
tel. 95/140047

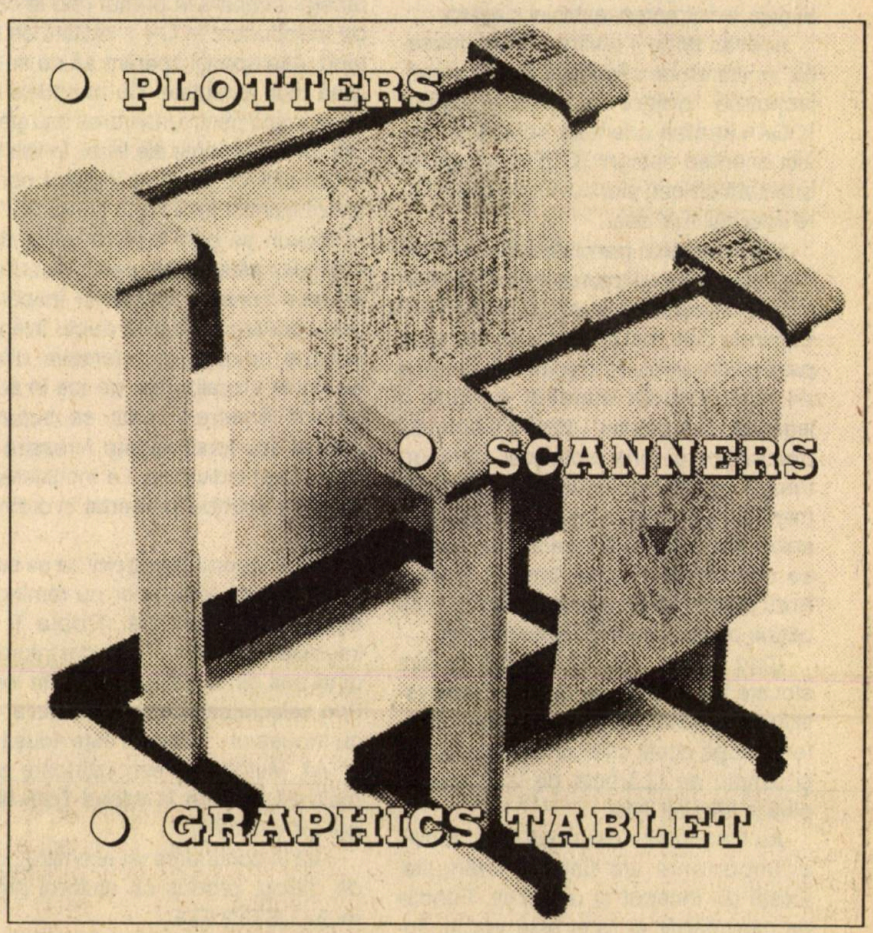
*** Gratis * Gratis * Gratis ***

scop

DISTRIBUITORUL AUTORIZAT
ÎN ROMÂNIA
DE PRODUSE ALE FIRMEI

Summagraphics / HOUSTON INSTRUMENT

Noi sîntem soluția



TELEFON: 11.74.21 / 11.92.48. FAX: 11.73.74.

Curs de programare OOP în C++, partea a patra

Stăpînind ierarhii

În precedentele 3 părţi aţi putut afla cele mai importante lucruri relative la C++. Vom consolida acum aceste cunoştinţe, pentru a le putea utiliza cu succes în practică.

Desigur, după ce aţi trecut prin primele 3 părţi ale acestui curs, nu mai este greu să definiţi clase proprii, împreună cu metodele adecvate. Funcţie de "antrenament", va fi posibilă curînd şi manevrarea lejeră a obiectelor dinamice. Dar, cu acestea, de-abia se stăpînesc uneltele "normale". Ca şi în viaţa reală, importantă este acordarea reciprocă a unor clase diferite, cu cele mai diverse funcţii element şi interacţiuni. Iar acest acord ar trebui obţinut fără constrîngeri, fără forţări asupra claselor deja existente. De aceea relaţiile între diferitele clase derivate trebuie analizate deja la formarea ierarhiei claselor.

Acestei teme îi revine un loc deosebit în literatura informatică mai nouă. Noţiunile "object oriented analysis" (OOA=analiză orientată obiect) şi "object oriented design" (OOD=proiectare orientată obiect) sînt cuvintele cheie care apar cel mai des.

Vă prezentăm pentru început un mic exemplu, care utilizează metodele descrise în literatură. Listingul LISTA1.CPP cuprinde cele mai importante elemente pentru liste dublu înlănţuite. Într-o astfel de listă, fiecare element conţine o legătură - un pointer - înspre elementul precedent şi una spre cel care urmează. În afară de aceasta, fiecare element este prevăzut cu un atribut, care stabileşte o legătură între elementele ce se conformează unui anumit criteriu. Prin intermediul acestui atribut se obţine o sortare a elementelor listei.

Cînd datele unei liste înlănţuite sînt stocate într-o colecţie de date, este relativ uşor posibilă tipărirea datelor existente după acest criteriu de sortare. Pur şi simplu se tipăreşte de la începutul pînă la sfîrşitul listei.

Au fost numite aici două caracteristici importante ale listelor înlănţuite. Există un început şi un sfîrşit. Funcţie de care capăt al listei este vorba, lipseşte legătura spre predecesor sau succesor. Lipsa acestor legături este marcată de pointerul vid (NULL). Astfel de pointeri marchează faptul că legătura nu este "reală", că datele la care se referă nu există de fapt. La adresa "NULL" nu se pot găsi date, căci prima adresă începe la "1".

Obiectele »A1« pînă la »A5« sînt create dinamic. Deoarece în momentul creerii încă nu există succesor, la iniţializarea obiectului s-a renunţat la conţinutul corect al legăturilor spre predecesor şi succesor. În acest exemplu, nu acesta este obiectivul principal. Menţionăm însă aici că metode precum adăugarea, ştergerea şi respectiv inserarea unui element în listă sînt indispensabile pentru o administrare confortabilă a listelor înlănţuite. Pentru metode care merg puţin mai departe, facem trimitere în primul rînd la cartea de introducere în C++ editată de Borland. Alte apariţii sperăm să nu se lase prea mult aşteptate. Cu metodele descrise acolo pentru inserarea sau ştergerea unor elemente ale listei, în listing ar fi posibilă dotarea cu legături corecte ale obiectelor încă de la generare.

"Luxul" pe care îl oferă listele dublu înlănţuite este că ele pot fi parcurse în ambele sensuri, înainte şi înapoi. Să presupunem că o listă dublu înlănţuită conţine pointeri spre ferestre diferite. Atunci la vizualizarea "de jos în sus" a tuturor ferestrelor care se suprapun (parţial sau total) fiecare fereastră ar fi "deschisă" individual. La închiderea ferestrelor ar trebui procedat în ordine inversă.

Dar, în cadrul ferestrelor ce se suprapun, ordinea ferestrelor nu rămîne totdeauna cea iniţială. Poate fi mai interesantă la un moment dat informaţia dintr-una din ferestrele aflate "în teanc". Prin selectarea acesteia din taste - sau cu mouse-ul - fereastra este adusă sus, în vîrf. Metoda corespunzătoare a fost realizată, în mare, în »Move-To()« (listingul Liste2.CPP).

Iată în continuare un exemplu posibil de dialog, produs cu ajutorul programului LISTE2.EXE:

```
1 2 3 4 5
Tastati un numar intre 1 si 5: 3
3 1 2 4 5
Tastati un numar intre 1 si 5: 4
4 3 1 2 5
Tastati un numar intre 1 si 5: 2
2 4 3 1 5
Tastati un numar intre 1 si 5: 5
5 2 4 3 1
Tastati un numar intre 1 si 5: 5
5 2 4 3 1
Tastati un numar intre 1 si 5: 1
1 5 2 4 3
Tastati un numar intre 1 si 5: 0
1 5 2 4 3
```

Cifra cea mai din stînga reprezintă fereastra cea mai de sus. Succesiunea ferestrelor nu se schimbă dacă, din întîmplare, se selectează acea fereastră care este deja în vîrf. În exemplu, a fost selectată de două ori la rînd fereastra 5. Deşi în metoda »Move-To()« succesiunea se schimbă, în implementarea de faţă nu s-a făcut nici o modificare a legăturilor spre succesor sau predecesor. Din contră, au fost interschimbate numai conţinuturile ferestrelor. Acest lucru se întîmplă prin intermediul metodei »Schimb()«. Menţionăm aici că prin aceasta uzul metodei se restricţionează destul de mult. Explicităm puţin dificultăţile realizării schimbului:

- al treilea caz se autoelimină prin simplitate, căci dacă fereastra cea mai de sus este mutată sus, nu se întîmplă absolut nimic
- în primul caz se modifică legăturile predecesorului, ale fostului element aflat sus şi ale noului element aflat în vîrf
- iar al doilea caz implică modificarea a chiar patru elemente diferite.

În listingul LISTE2.CPP se tot schimbă două conţinuturi învecinate între ele, pînă cînd elementul dorit se află în vîrf. Această metodă are dezavantajul că, la liste lungi, acest lucru va dura desigur foarte mult.

În listingul FEREAST1.CPP sînt realizate metodele fundamentale pentru reprezentarea unei ferestre simple. Aceste metode se aseamănă cu cele pe care le-am prezentat în listingurile

anterioare, relative la dreptunghiuri şi pătrate. La acele listinguri ne putem imagina şi funcţii element care să influenţeze dimensiunile şi poziţia ferestrei. De aceea, memoria tampon pentru o fereastră a fost dimensionată astfel încît - la operaţiuni "zoom" de exemplu - să fie posibilă reprezentarea ferestrei pe întregul ecran.

Pentru fiecare fereastră, coordonatele poziţiei, lăţimea, înălţimea precum şi textul corespunzător pot fi uşor accedate. Fiecare obiect fereastră poate fi vizualizat pe ecran cu ajutorul numărului său intern.

Listingul FEREAST1.CPP nu urmăreşte poziţia relativă a diverselor ferestre între ele. Nu este posibil aici un »refresh« (o reîmprospătare) completă a ferestrelor "de jos în sus". Pe ecran, obiecte tridimensionale sînt reprezentate într-o suprafaţă bidimensională. Să presupunem că o fereastră oarecare se translatează sau este redimensionată de aşa manieră încît devine necesar un "refresh" al reprezentării. Atunci, din păcate, colecţia de metode din listingul FEREAST1.CPP se dovedeşte insuficientă. De aceea, a fost realizat listingul FEREAST2.CPP.

Dar, înainte, să menţionăm şi o altă posibilitate de rezolvare. Proprietăţile clasei »Lista« în listingul LISTE1.CPP şi metodele clasei FEREAST1.CPP sînt de fapt suficiente pentru a realiza o legătură prin tehnici de programare uzuale. Succesiunea ferestrelor s-ar păstra în clasa »Lista«. Ferestrele însele sînt conţinute în clasa »Fereast«. Cu funcţia »Get-Elem()« s-ar putea returna o valoare întreagă care să specifice poziţia în clasa ferestrelor. Dar, în spiritul încapsulării datelor, cu noile metode este mai avantajoasă derivarea unei clase noi, din clasele »Fereast« şi »Lista«. În listingul »Fereast2.CPP«, această clasă derivată se numeşte »Vedere«. În clasa »Vedere« sînt reunite metodele ambelor clase. O instanţă a acestei clase are în antet 3 pointeri. Cei doi pointeri »prev« şi »succ« sînt legăturile cunoscute spre predecesor şi succesor în cadrul listei. Dar, în locul unei valori numerice a elementului listei, spre deosebire de listingurile LISTE1.CPP şi LISTE2.CPP, aici s-a folosit un al 3-lea pointer spre elementul corespunzător. Acest pointer este mai uşor modificabil decît o cantitate mai mare de elemente-date, cum apăreau în fosta clasă »Fereastr«.

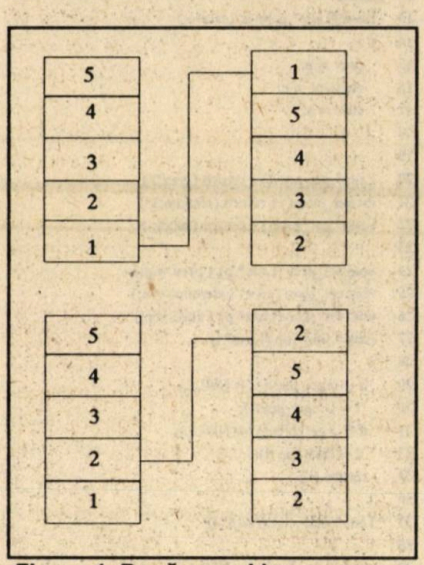


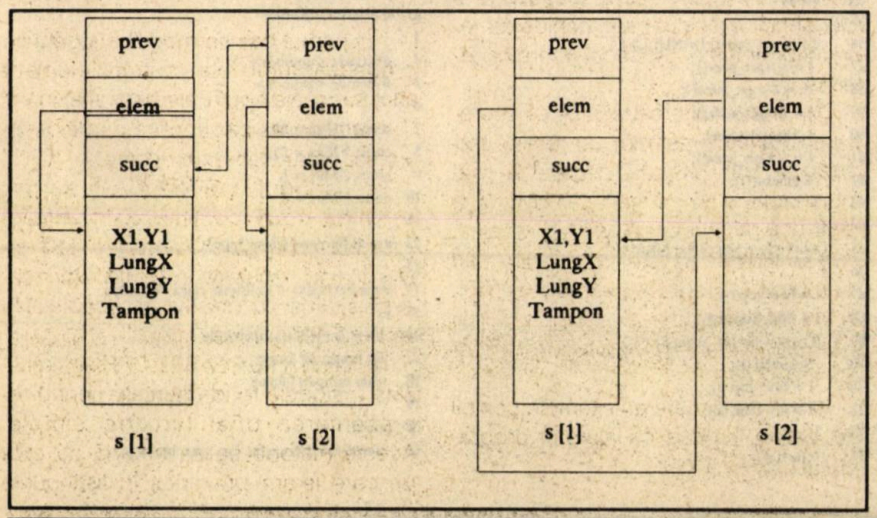
Figura 1. Două cazuri la translatarea ferestrelor suprapuse

În figura 2, partea superioară provine din clasa »Liste« (3 pointeri), iar partea inferioară este analogă clasei »Fereastr«. În metoda »Move-Top()«, care are ca efect deplasarea celei de-a n-a ferestre (numărînd de sus), nu trebuie schimbate părţile inferioare ale instanţelor. Este suficient să se interschimbe pointerii spre ferestrele corespunzătoare. Este mult mai rapidă modificarea unui pointer decît mutarea unui domeniu la care face referinţă acest pointer. În ultimul caz, domeniul de memorie care ar trebui mutat este semnificativ mai mare. Şi în sensul unei valabilităţi generale, este mai bine să se acţioneze numai asupra pointerilor, deoarece atunci aceştia pot face referire la obiecte de dimensiuni diferite.

Metoda »Fereastr::Afis()« face uz de datele din partea inferioară fără a folosi pointerul spre elementul corespunzător. În schimb metoda »View::Afis()« foloseşte pointerul spre fereastra eventual modificată a unei instanţe învecinate a clasei »View«. Din acest motiv a fost necesară formularea funcţiilor element specifice clasei »Fereastr« ca funcţii virtuale. În programul FEREAST2.EXE, la tastarea valorii »1« vederea spre ferestre nu se schimbă în nici un fel, deoarece fereastra cea mai de sus rămîne sus. Dacă se alege de cîteva ori la rînd cifra »2«, apar alternativ cele două ferestre aflate sus (se modifică aspectul). Acest lucru este reprezentat schematic în figura 2. La alegerea cifrei »5«, fereastra cea mai de jos (a 5-a numărînd de sus) va fi adusă în vîrf. Fosta fereastră cu numărul 4 devine noua fereastră 5 şi deci se va găsi la baza "stivei" de ferestre.

Listingul FEREAST2.CPP e gîndit ca o sugestie pentru integrarea altor metode pentru ferestre suprapuse. Intenţionat am renunţat la o colecţie cuprinzătoare de metode deja reprezentate. Programarea orientată obiect poate fi învăţată aproape în joacă, prin încercarea de multiple combinaţii a unor clase diferite şi ale metodelor lor. Cititorului îi va face cu siguranţă mai

Figura 2. Doua instanţe schematice ale clasei "Vedere", în starea iniţială şi după interschimbarea celor două ferestre superioare



multă plăcere să creeze o metodă nouă, decît să tasteze pînă la epuizare, introducînd un listing fără sfîrşit. OOP-ul este şi mai uşor înţeles în acest fel. În acest sens: mult succes în dezvoltarea unor metode noi!

(J.S./I.F.)

```
1   // Lista1.cpp
2   // lista inlantuita
3
4   #include <conio.h>
5   #include <iostream.h>
6
7   class Lista
8   {
9    Lista *prev;
10   int   element;
11   Lista *succ;
12
13  public:
14   Lista (Lista* p, int e, Lista* s)
15   {
16    prev = p;
17    element = e;
18    succ = s;
19   };
20
21   Lista* get_prev() { return (prev); }
22   int get_elem() { return (element); }
23   Lista* get_succ() { return (succ); }
24
25   void set_prev (Lista* p) { prev = p;}
26   void set_elem (int e) { element = e;}
27   void set_succ (Lista* p) { succ = p;}
28  };
29
30  int main()
31  {
32   clrscr();
33
34   Lista* A1 = new Lista (NULL, 1, NULL);
35   Lista* A2 = new Lista (A1, 2, NULL);
36   A1->set_succ (A2);
37   Lista* A3 = new Lista (A2, 3, NULL);
38   A2->set_succ (A3);
39   Lista* A4 = new Lista (A3, 4, NULL);
40   A3->set_succ (A4);
41   Lista* A5 = new Lista (A4, 5, NULL);
42   A4->set_succ (A5);
43
44   Lista* A0 = new Lista (*A1);
45   cout <<"A1: " << A0->get_elem() << "\n";
46
47   A0 = A0->get_succ();
48   cout << "A2: " << A0->get_elem() << "\n";
49
50   A0 = A0->get_succ();
51   cout << "A3: " << A0->get_elem() << "\n";
52
53   A0 = A0->get_succ();
54   cout << "A4: " << A0->get_elem() << "\n";
55
56   A0 = A0->get_succ();
57   cout << "A5: " << A0->get_elem() << "\n";
58  }
```

LISTA1.CPP

```
1   // Lista2.cpp
2   // lista inlantuita
3
4   #include <conio.h>
5   #include <iostream.h>
6
7   class Lista
8   {
9    Lista *prev;
10   int   element;
11   Lista *succ;
12  public:
13   Lista (Lista* p, int e, Lista* s)
14   {
15    prev = p;
16    element = e;
17    succ = s;
18   }
19
20   Lista* get_prev() { return (prev); }
21   int get_elem() { return (element); }
22   Lista* get_succ() { return (succ); }
23
24   void set_prev (Lista* p) { prev = p;}
25   void set_elem (int e) { element = e;}
26   void set_succ (Lista* p) { succ = p;}
27   Lista* find_top (Lista* l)
28   {
29    if (l->get_prev() != NULL)
30     l = l->get_prev();
31    if (l->get_prev() != NULL)
32     l = find_top (l);
33    return (l);
34   }
35   Lista* find_end (Lista* l)
36   {
37    if (l->get_succ() != NULL)
38     l = l->get_succ();
39    if (l->get_succ() != NULL)
40     l = find_end (l);
41    return (l);
42   }
43   Lista* find_next (int e, Lista* l)
44   {
45    if (l->get_elem() != e)
46     l = l->get_succ();
47    if (l->get_elem() != e)
48     if (l->get_succ() != NULL)
49      l = find_next (e, l);
50    return (l);
51   }
52   void Schimba (Lista* l)
53   {
54    int m;
55
56    if (l->get_prev() != NULL) {
57     l = l->get_prev();
58     m = l->get_elem();
59     l = l->get_succ();
60     l->set_elem(m);
61     l = l->get_prev();
62     Schimba (l);
63    }
64   }
65   Lista* move_top (int e, Lista* l)
66   {
67    l = find_top (l);
68    l = find_next (e,l);
69    if (e == l->get_elem() ) {
70     Schimba (l);
71     l = find_top (l);
72     l->set_elem (e);
73    }
74    return (l);
75   }
76   void Afisare (Lista* l)
77   {
78    cout << " " << l->get_elem() << " ";
79    if (l->get_succ() != NULL) {
80     l = l->get_succ();
81     Afisare (l);
82    }
83    else cout << "\n";
84   }
85   void Prezinta (Lista* l)
86   {
87    l = find_top (l);
88    Afisare (l);
89   }
90  };
91
92  main()
93  {
94   int t;
95
96   clrscr();
97   Lista* A1 = new Lista (NULL, 1, NULL);
98   Lista* A2 = new Lista (A1, 2, NULL);
99   A1->set_succ (A2);
100  Lista* A3 = new Lista (A2, 3, NULL);
101  A2->set_succ (A3);
102  Lista* A4 = new Lista (A3, 4, NULL);
103  A3->set_succ (A4);
104  Lista* A5 = new Lista (A4, 5, NULL);
105  A4->set_succ (A5);
106
107  Lista* A0 = new Lista (*A1);
108  A0->Prezinta (A0);
109
110  t = 1;
111  while (t > 0) {
112   cout << "Tastati un numar intre 1..5 (0 termina): ";
113   cin >> t;
114   if (t <= 5) {
115    A0 = A0->move_top (t, A0);
116    A0->Prezinta (A0);
117   }
118  }
119 }
```

LISTA2.CPP

```
1   // Fereast1.cpp
2   // ferestre dinamice
3
4   #include <conio.h>
5   #include <stdio.h>
6
7   const XMax = 80;
8   const YMax = 25;
9   const XMin = 1;
10  const YMin = 2;
11
12  enum Boolean {false, true};
13
14  class Fereastra  // defineste clasa Fereastra
15  {
16   int x;  int y; // coordonatele
17   int lungx; int lungy;
18   char tampon [4096];
19
20  public:
21   Fereastra::Fereastra (int initx, int inity, \
```

DYNA-STATION 386DS20/25	
CPU	80386SX-20/25
Viteza	20 sau 25 MHz
RAM (standard-max.)	1 MB - 8 MB, SIMM
BIOS	64 KB
Coprocesor (opţional)	80387SX
Porturi seriale/paralele	1 serial/ 1 paralel
Conector extern FDD	da (opţ. al 2-lea port serial sau game port)
Port mouse PS/2	da
Controller floppy disc	1,44 Mb/ 720 kb/ 1,2 Mb/ 360 Kb
Controller hard disk	IDE
Lector de dischete 3,5"	1 x 1,44 Mb
Hard disk 3,5' (opţional)	40/ 80/ 120/ 210 Mb
Sloturi de extensie, 16 biţi	1 slot (dacă nu este instalat HDD)
Versiuni VGA	256 Kb video RAM: 640x480x16, 800x600x16 culori (Trident TVGA 9000) 512 Kb video RAM: 640x480x256 (non interlaced), 800x600x256 (non-interlaced), 1024x768x16 culori
Dimensiuni	28,5 x 5 x 22 cm
Greutate	2,5 Kg
Sisteme de operare	MS-DOS 5.0 sau 4.01, OS/2, Windows 3.0, Unix, Novell Netware, OS/2 LAN Manager

LECOM S.R.L., Calea Victoriei nr. 126, Bucureşti, tel. 506420, fax: 129558

```
                        int lx,  int ly)
{
 x = initx; lungx = lx - x;
 y = inity; lungy = ly - y;
}
// functii element publice
int Fereastra::GetX1() { return x; }
int Fereastra::GetY1() { return y; }
int Fereastra::GetX2() { return (x+lungx);}
int Fereastra::GetY2() { return (y+lungy);}
int Fereastra::GetLX() { return lungx;}
int Fereastra::GetLY() { return lungy;}
void Fereastra::Citire()
{
 gettext (GetX1(), GetY1(), GetX2(), \
                        GetY2(), &tampon);
}
void Fereastra::Afisare()
{
 puttext (GetX1(), GetY1(), GetX2(), \
                        GetY2(), &tampon);
}
void Fereastra::Umplere (char valoare, char atr)
{
 int i;

 for (i = 0; i < 4096; i = i+2) {
    tampon[i] = valoare; tampon[i+1] = atr;
  }
}
};

int main()
{
 char tasta;
 Boolean sfirsit;

 clrscr();
 Fereastra *F1 = new Fereastra (XMin, YMin, \
                        XMax, YMax);
 F1->Umplere ('1',0x16);
 F1->Afisare();

 Fereastra *F2 = new Fereastra (5, 3, \

 F2->Umplere ('2', 0x2f);
 F2->Afisare();

 Fereastra *F3 = new Fereastra (8, 5, 10, 20);
 F3->Umplere ('3', 0x38);
 F3->Afisare();

 Fereastra *F4 = new Fereastra (16, 12, 26, 14);
 F4->Umplere ('4', 0x40);
 F4->Afisare();

 Fereastra *F5 = new Fereastra (36, 18, 76, 22);
 F5->Umplere ('5', 0x51);
 F5->Afisare();

 Fereastra *F6 = new Fereastra (50, 15, 62, 19);
 F6->Umplere ('6', 0x6a);
 F6->Afisare();

 printf ("Alegeti un numar intre 1..6: ");
 do {
  tasta = getch ();
  sfirsit = false;
  switch (tasta)
   {
    case '1': F1->Afisare();
              break;
    case '2': F2->Afisare();
              break;
    case '3': F3->Afisare();
              break;
    case '4': F4->Afisare();
              break;
    case '5': F5->Afisare();
              break;
    case '6': F6->Afisare();
              break;
    default: sfirsit = true;
   };
 } while (!sfirsit);
}
```

FEREAST1.CPP

```
// Fereast2.cpp
// ferestre dinamice in liste inlantuite

#include <conio.h>
#include <stdio.h>

const XMax = 80;
const YMax = 25;
const XMin = 1;
const YMin = 2;

enum Boolean {false, true};

class Fereastra  // defineste clasa Fereastra
{
 int x; int y; // coordonatele
 int lungx; int lungy;
 char tampon [4096];

public:
 Fereastra::Fereastra (int initx, int inity, \
                        int lx,  int ly)
 {
  x = initx; lungx = lx - x;
  y = inity; lungy = ly - y;
 }
 // functii element publice
 int Fereastra::GetX1() { return x; }
 int Fereastra::GetY1() { return y; }
 int Fereastra::GetX2() { return (x+lungx);}
 int Fereastra::GetY2() { return (y+lungy);}
 int Fereastra::GetLX() { return lungx;}
 int Fereastra::GetLY() { return lungy;}
 char* Fereastra::GetBuf() { return tampon;}
 void Fereastra::Citire()
 {
  gettext (GetX1(), GetY1(), GetX2(), \
                        GetY2(), &tampon);
 }
 virtual void Fereastra::Afisare()
 {
  puttext (GetX1(), GetY1(), GetX2(), \
                        GetY2(), &tampon);
 }
 virtual
 void Fereastra::Umplere (char valoare, char atr)
 {
  int i;

  for (i = 0; i < 4096; i = i+2) {
     tampon[i] = valoare; tampon[i+1] = atr;
   }
  }
};

class Lista
{
 Lista *prev;
 Fereastra *element;
 Lista *succ;
public:
 Lista::Lista (Lista *p = NULL, \
               Fereastra *e = NULL, \
               Lista *s = NULL)
 { prev = p; element = e; succ = s; }

 Lista* get_prev() { return (prev); }
 Fereastra* get_elem() { return (element); }
 Lista* get_succ() { return (succ); }

 void set_prev (Lista* p) { prev = p;}
 void set_elem (Fereastra* e) { element = e;}
 void set_succ (Lista* p) { succ = p;}
 void set_list (Lista* p, Fereastra* e, Lista* s)
 { prev = p; element = e; succ = s;};
 Lista* find_top (Lista* l)
 {
  if (l->get_prev() != NULL)
   l = l->get_prev();
  if (l->get_prev() != NULL)
   l = find_top (l);
  return (l);
 }
 Lista* find_end (Lista* l)
 {
  if (l->get_succ() != NULL)
   l = l->get_succ();
  if (l->get_succ() != NULL)
   l = find_end (l);
  return (l);
 }
 Lista* find_next (int e, Lista* l)
 {
  int z;
  for (z = 1; z < e; z++) {
   if (l->get_succ() != NULL)
    l = l->get_succ();
  }
  return (l);
 }
 void Schimba (Lista* l)
 {
  Fereastra* m; Fereastra* n;

  if (l->get_prev() != NULL) {
   n = l->get_elem();
   l = l->get_prev();
   m = l->get_elem();
   l->set_elem(n);
   l = l->get_succ();
   l->set_elem(m);
   l = l->get_prev();
   Schimba (l);
  }
 }
 Lista* move_top (int e, Lista* l)
 {
  l = find_top (l);
  l = find_next (e,l);
  Schimba (l);
  return (l);
 }
};

class View : public Fereastra, public Lista
{
 int vizibil;
```

(continuare în pag. 62)

Programarea Windows (I)

Ca mediu de programare pentru Windows s-ar fi putut folosi mai multe compilatoare C, dar s-a depins de SDK şi WRT. Acest lucru s-a schimbat radical odată cu apariţia noilor produse. Această serie de articole va arăta cum funcţionează Windows şi cum se poate programa: simplu sau complicat.

Pentru utilizatorii DOS, Windows 3.0 este o suprafaţă utilizator grafică foarte frumoasă. Pentru impunerea sa pe piaţă hotărîtoare a fost disponibilizarea rapidă a aplicaţiilor şi rolul conducător al firmei Microsoft pe piaţă. Împotriva acestora GEOWORKS Ensemble nu a putut concura cu toate că este o suprafaţă mai bună şi mai rapidă.

La acestea se mai adaugă faptul că Microsoft însăşi a extras unele produse din versiunea 2 şi le-a împins cu forţă în regimul versiunii superioare. Gigantul software şi-a dat seama că Windows se va putea vinde mai bine dacă se va face multă vîlvă în jurul produsului, de aceea mai întîi se oferă cantitate, abia apoi urmîndu-se a se ajunge şi la calitate.

Se pare că Borland a cîştigat prima rundă în lupta pentru limbajele de programare aferente Windows. Producătorul de software, proaspăt "căsătorit" cu Ashton Tate nu a făcut, nici mai mult nici mai puţin, decît să-şi redezvolte toate produsele cunoscute de pînă acum, dar rezultatul a fost mai bun calitativ decît SDK-ul sau WRT-ul (Whitewater Resource Toolkit) lui Microsoft. Pe lîngă acest fapt produsele Borland sînt mult mai ieftine. (Concurenţa este mult mai înrăită în SUA!). Firma Borland oferă produse calitativ superioare la jumătatea preţului celor de la Microsoft.

Făcînd o comparaţie între MS-C 6.0, (între timp a apărut versiunea 7.0), şi SDK cu C++ şi Object Windows Library ale firmei Borland, se ajunge repede la concluzia, că programarea orientată obiect (OOP) este tehnica cea mai comodă de scriere a aplicaţiilor Windows.

Pentru a putea trage asemenea concluzii este necesară cunoaşterea şi înţelegerea mecanismelor interne Windows.

Faceţi cunoştinţă cu Windows

Programele "clasice" sînt interpretate secvenţial: Interpretorul de comenzi poziţionează contorul de comenzi al procesorului la începutul programului, iar apoi acesta prelucrează comandă după comandă. Există şi un sfîrşit definit. Derularea programului este definită de structura programului. Cei care cunosc această structură o denumesc "utilă" iar cei mai conservatori o consideră singura cale spre o adevărată programare.

Programele Windows sînt puse în mişcare de informaţii, deci ele funcţionează cu totul diferit decît programele secvenţiale. Fiecare aplicaţie este compusă dintr-o multitudine de componente singulare, fiecare dintre acestea reprezentînd o funcţie bine definită. Aici se crează informaţii fie printr-o funcţie parţială a programului, fie printr-un eveniment din afară (clic pe mouse, apăsarea pe o tastă, starea imprimantei etc.). Acestea trebuie să provoace o anumită reacţie a funcţiilor de care aparţine. De aceea nu se poate determina cu exactitate modul de derulare al unui astfel de program.

Dacă, spre exemplu cursorul mouse-ului s-a poziţionat undeva pe ecran şi a fost apăsată o tastă a mouse-ului, un handler de evenimente determină cînd şi unde s-a întîmplat, găseşte funcţia de care aparţine şi îi predă mesajul corespunzător. Funcţia reacţionează la informaţie în mod individual şi transmite programului părinte un mesaj de terminare (rezolvare). Acest mecanism se repetă într-o buclă infinită.

Ieşirile pe ecran generate de toate programele Windows sînt de natură grafică.

Acest fapt cere un cu totul alt mod de gîndire, mai ales pentru cei care au programat pînă acum orientat caracter. Ieşirile grafice pot fi puncte singulare, cercuri, dreptunghiuri, linii sau poligoane. Dar şi afişarea textelor se face în mod grafic. Astfel s-a uşurat folosirea comună în aplicaţii a textului cu grafica. Tratarea textului pierde din confort, dar avantajele obţinute merită osteneala.

Pentru ca utilizatorul să se ocupe cît mai puţin de detaliile interne ale ieşirilor grafice, Windows îi pune la dispoziţie un GDI (Graphic Developement Interface). Cu aceasta pot fi reprezentate elementele grafice cele mai utilizate. Graficele obţinute pot fi dependente sau independente de contextul ieşirii. Deci cu acelaşi apel al funcţiei "Rectangle" se poate afişa un dreptunghi atît pe ecran cît şi la imprimantă. Din punctul de vedere al utilizatorului ieşirea pe ecran poate fi identică cu cea de la imprimantă (WYSIWYG).

Pentru a putea realiza această unitate Windows necesită unităţi inteligente, care pot reprezenta toate elementele susţinute de GDI. Aceste unităţi se denumes driver-e. Ele declanşează ieşirea grafică propriu-zisă. GDI-ul recunoaşte patru unităţi: ecranul, aparate hardcopy, cum ar fi imprimanta şi plotter-ul, bitmap-uri şi metafişiere.

Ecranele şi imprimantele sînt echipamente fizice, care redau imaginile în mod optic. Celelalte două: bitmap-urile şi metafişierele sînt pseudounităţi. Ele permit memorarea imaginilor în memoria de lucru sau pe harddisk.

Oricărei aplicaţii de sub Windows îi este subordonată o fereastră grafică. Ea constă din două zone distincte: zona non-client, care este gestionată de obicei de sistem şi poate cuprinde un chenar activ (border resizeble), care prin "tragere" rescalează fereastra, precum şi un antet pe care e afişat de obicei numele aplicaţiei şi de care dacă se "trage" se mută fereastra aplicaţiei oriunde pe ecran. Acest antet mai poate cuprinde trei controale specifice - Sistem meniu (un meniu scurt cu funcţii de interfaţă cu Windows), buton de maximizare (fereastra va acoperi tot ecranul) şi buton de minimizare (fereastra revine la mărimea normală dacă era maximizată, sau devine icoană - o reprezentare grafică restrînsă a aplicaţiei). Restul suprafeţei ferestrei se numeşte zonă client şi aici se manifestă cu precădere funcţiile specifice aplicaţiei. Originea sistemului de coordonate este definit astfel încît în colţul din stînga sus să se afle domeniul client. Concomitent un mecanism de siguranţă veghează, ca fereastra să nu încalce o altă fereastră şi invers. Înseamnă că obiectele grafice care depăşesc marginea ferestrei vor fi tăiate automat.

```
#include <Windows.h>

long FAR PASCAL FirstWndProc (HWND,unsigned,WORD,LONG);

int PASCAL WinMain      (HANDLE    hInstance,
                         HANDLE    hPrevInstance,
                         LPSTR     lpszCmdLine,
                         int       cmdShow)
{
HWND                    hwnd;
MSG                     msg;
WNDCLASS                wndclass;

if (!hPrevInstance)
  {
   wndclass.lpszClassName       = "FIRST:MAIN";
   wndclass.hInstance           = hInstance;
   wndclass.lpfnWndProc         = FirstWndProc;
   wndclass.hCursor             = LoadCursor(hInstance,"Sageata");
   wndclass.hIcon               = LoadIcon (hInstance,"Icon_1");
   wndclass.lpszMenuName        = NULL;
   wndclass.hbrBackGround       = COLOR_WINDOW+1;
   wndclass.style               = NULL;
   wndclass.cbClsExtra          = 0;
   wndclass.cbWndExtra          = 0;
   RegisterClass(&wndclass);
  }
hwnd = CreateWindow     ("FIRST:MAIN",
                         "Meniu principal",
                         WS_OVERLAPPEDWINDOW,
                         CW_USEDEFAULT,
                         0,
                         CW_USEDEFAULT,
                         NULL,
                         NULL,
                         hInstance,

                         NULL);
ShowWindow (hwnd,cmdShow);
while (GetMessage (&msg,0,0,0))
  {
   TranslateMessage (&msg);
   DispatchMessage (&msg);
  }
return 0;

long FAR PASCAL FirstWndProc  (HWND     hwnd,
                               unsigned msg,
                               WORD     wParam,
                               LONG     lParam)
{
switch (msg)
  {
   case WM_DESTROY:
        PostQuitMessage(0);
        break;
   default:
        return (DefWindowProc(hWnd,msg,wParam,lParam));
        break;
  }
return 0L;
}
}
```

Fişierul FIRST.C - Primul program C pentru Windows

```
NAME FIRST

EXETYPE WINDOWS

DESCIPTION 'Primul inceput'

CODE MOVEABLE DISCARDABLE
DATA MOVEABLE MULTIPLE

HEAPSIZE 512

STACKSIZE 5000

EXPORTS
    FirstWndProc
```

Fişierul FIRST.DEF - Fişierul de definire a modulelor

```
first.exe: first.obj first.def first.res
  link first,first/align:16,first/map,mlibcew libw/NOD/NOE/CO, min.def
  rc min.res

first.res: first.rc first.crs first.ico
  rc -r first.rc

first.obj: first.c
  cl -AM -c -Gsw -Od -W2 -Zpi first.c
```

Fişierul FIRST.MAK - Fişierul MAKE.

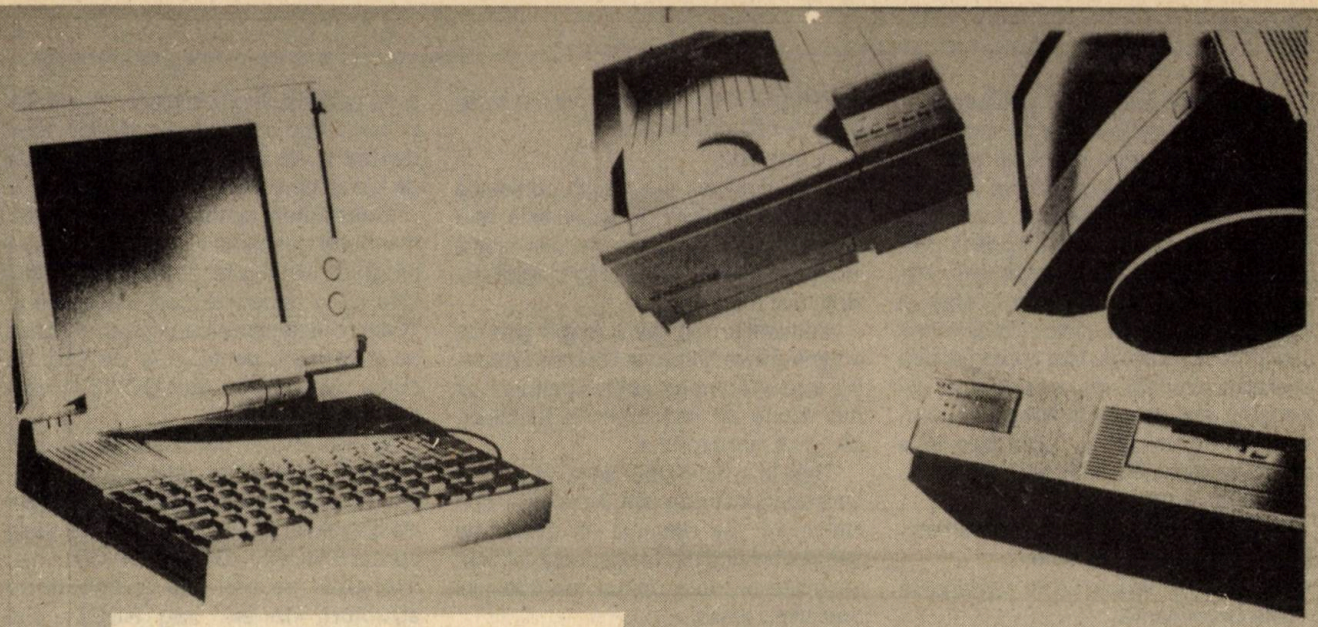
```
Icon_1      icon        first.ico
Sageata     cursor      first.crs
```

Fişierul FIRST.RC - Fişierul de resurse.

Listing 1.1. Scheletul de bază al unui program Windows scris în limbajul MS-C.

```
Program First;
{$R a3.res}
Uses WObjects;

type

   MyApp = Object (TApplication)
         Procedure InitMainWindow;Virtual;
   End;

Procedure MyApp.InitMainWindow;
Begin
   MainWindow := New(PMenuWindow, Init);
End;

Var MyA : MyApp;

Begin
   MyA.Init('First');
   MyA.Run;
   MyA.Done;
End.
```

Listing 1.2. Orientat obiect cu Turbo-Pascal for Windows.



NEC - UltraLite SL/20P

Procesor: Intel 386SL-20, 20 MHz, comutabil la 10 MHz; soclu pentru coprocesor Intel 80387SX; ROM 128 KB; memoria de lucru 2 MB (80 ns), extensibilă cu 2, sau 6 MB, pînă la max. 8 MB; cache 16 KB; Power Management: în microprocesor; monitor: LCD Touch Screen pentru operare cu creion, diagonala 9,4", 25 x 80 caractere, supertwisted nematic (STN), cold cathodefluroreszent (CCFT), sidelit LCD; Video-RAM: VGA 640 x 480 puncte, 256 KB video-RAM, reprezentare în 16/32 nuanţe de gri; monitor extern: port VGA pentru cuplarea unui monitor extern cu o rezoluţie de 1024 x 768; hard disk 80 MB (IDE 19ms); floppy: 3,5" (1,44 MB) intern; interfeţe: conector video VGA, 1 serială (RS232C), 1 paralelă (Centronics), IDE pe motherboard, floppy extern sau Docking Station, încărcare baterie sau baterie externă, conector pentru bloc numeric extern, conector pentru tastatură şi mouse, Pen port; slot pentru cartelă fax/modem; tastatura: 79 taste, 12 taste funcţionale; Dimensiuni: 292x55x230 mm; Masa: 3,1 kg.

NEC Maus optional

NEC - ProSpeed 486SX/C

Procesor: Intel i486SX-20, 8 KB cache intern; 20 MHz; soclu pentru coprocesor i487SX; ROM: 64 KB sistem/32 KB video; memoria de lucru: 2-MB (80 ns) extensibilă pînă la max. 20/32 MB; monitor: 10,4" TFT color monitor cu matrice activă, VGA 640x480 puncte, 256 din 4096 de culori; video RAM: 512 KB; port pentru monitor VGA extern; hard disk: 120 MB (IDE, 19ms); floppy: 3,5" (1,44MB) intern; interfeţe: un slot 8/16/32 biţi EISA, IDE pe motherboard, conector video VGA, 1 serială, 1 paralelă, floppy extern, conector mouse; slot pentru cartelă fax/modem; tastatura: 92 taste, 12 taste funcţionale; bloc numeric separat; dimensiuni: 375x106x396 mm, masa: 7,6 kg.

LECOM

Distribuitor autorizat NEC:
LECOM S.R.L.
Calea Victoriei nr. 126
70179 Bucureşti
tel.: 50.64.20; 50.79.94; 12.95.58
fax: 12.95.58

Obiectele interfeţei utilizator

Este vorba despre obiecte care marchează pe ecran un domeniu din care utilizatorul poate să-şi rezolve intrările. Ele pot fi: ferestre, meniuri, icon-uri şi boxe de dialog. Windows sprijină programatorul cu funcţiuni predefinite, astfel încît efortul pentru producerea şi întreţinerea lor să fie minim. Trebuie însă ştiut modul în care au fost implementate obiectele interfeţei utilizator. În plus, importantă mai este şi cunoaşterea asocierii informaţiilor cu obiectele şi a anumitor rutine de bibliotecă Windows cu controlul obiectelor.

Aplicaţiile Windows pun programatorilor următoarele probleme:

- 1. Aplicaţiile trebuie să fie comandate prin evenimente
- 2. Windows cere ca toate ieşirile spre periferice (ecran, imprimantă) să fie de natură grafică
- 3. Folosire consecventă a obiectelor interfeţei utilizator

Limbajul este important

În trecut SDK era singura bază a programării eficiente Windows. Apoi s-au mai folosit şi compilatoare C compatibile Windows. În acest scop se pretează pe lîngă produsul Microsoft şi compilatoarele firmelor Jensen Partner MetaWare şi Zortech.

Problema în cazul programării Windows îl constituie faptul că Windows funcţionează comandat de evenimente, prin schimbul de informaţii dintre diferitele obiecte, el însuşi nefiind însă programat orientat obiect.

De aceea programatorii refuză să introducă compilatorul C++ compatibil Windows în speranţa unei simplificări a scrierii de aplicaţii.

Motivul este clar. Proiectanţii firmei Microsoft au prezentat o cale de mijloc între o arhitectură "precedentă" şi una comandată prin evenimente, deoarece compilatorul C este capabil să lucreze cu arhitectura modernă, doar cu un efort foarte mare. Conflictul apărut între C++ şi Windows este comparabil cu contrastul dintre soluţii simple şi probleme complicate.

Borland vrea să înlăture această anomalie. Cele două vîrfuri ale sale, C++ şi Pascal orientat obiect, se livrează deja pentru utilizare sub Windows. Cu ajutorul unei aplicaţii minimale se va putea arăta că OOP simplifică cu mult programarea. O bibliotecă de clase cu structurile sale tipice de ereditate poate fi manipulată mult mai uşor decît o culegere de 600 de funcţii API.

Convenţii

Sub Windows s-a impus convenţia de nume maghiară. Scopul ei este de a face codul de program mai clar, astfel întreţinerea ulterioară a programelor devine mult mai simplă.

Convenţia maghiară, (după ţara de origine a inventatorului Charles Simonyi), este un compromis între denumiri cît mai scurte ale variabilelor şi funcţiilor, dar şi cît mai explicite.

Tabelul 1 ne arată o serie de prefixe care apar destul de des. Astfel prin "ach Name 14" s-a denumit un sector de semn, constînd din 14 elemente, iar "ich" se foloseşte pentru denumirea indexului unui cîmp grafic.

Pentru identificarea obiectelor, Windows foloseşte cifre de tip Integer fără semn, avînd o lungime de 16 biţi. Un astfel de număr se mai numeşte şi Handle. Singurul lor scop este identificare, iar o conversie în pointer nu dă nici o informaţie utilă. Unele funcţii API returnează un Handle. Faţă de DOS un NULL-Handle sub Windows este un Handle anulat, iar un Handle de fişier anulat are valoarea "-1", deoarece DOS distribuie Handle-re de fişiere, şi nu suprafaţa Windows.

Fişierul de header Windows

Orice aplicaţie Windows necesită o referinţă asupra fişierului WINDOWS.H. Acest fişier este foarte mare (110 KByte) şi nu conţine decît definiţiile constantelor simbolice ale celor mai diferite tipuri de date şi ale aşa-numitelor prototipuri de funcţii bibliotecă.

Constantele simbolice folosesc unei mai bune descifrări şi întreţineri a unui program. În cazul lor se înlocuiesc nişte cifre cu o denumire corespunzătoare. Toate constantele simbolice apar scrise cu litere mari, pentru a le deosebi de nume de variabile (WINDOWS.H defineşte în jur de 1500).

WINDOWS.H defineşte şi o serie întreagă de tipuri de date. Ele au menirea de a mări volumul de tipuri de date ale limbajului C. Pe lîngă tipurile uzuale cum ar fi "BOOL", "BYTE", "DWORD" sau "HANDLE" se mai află şi definiţii de structuri, de ex: "tag RECT". Ele uşurează referirea la diferitele tipuri de date, care aparţin unui obiect (coordonatele colţurilor unui dreptunghi).

Au fost definite prototipuri de funcţii de header în fişierul Windows pentru a putea observa deja în timpul compilării, dacă rutinele au fost apelate cu un număr eronat de parametri, cu tipuri greşite de parametri sau cu valori de retur eronate. Ele indică cum ar trebui să arate apelul unei rutine.

Mesaje

S-a mai spus că programele Windows folosesc un model de programare condus prin mesaje. Deci nu există o ordine prescrisă de derulare a unui pro-

Prefix	Tip date	Definiţie WINDOWS.H
a	Array (cîmp)	-
ch	Caracter (character)	-
cb	Număr de octeţi (count of Bytes)	-
dw	double word (unsigned long)	DWORD
h	Handle	-
hdc	Handle la contextul unităţii	-
hwnd	Handle la o fereastră	-
i	Index (întotdeauna compus)	-
l	long Integer	LONG
lp	long Pointer (tip compus far)	-
n	Integer	-
np	near Pointer	-
pt	Point (Structură Punct - x,y)	POINT
r	Rectangle (Structură dreptunghiulară)	RECT
sz	Null terminated string	-
w	word (unsigned integer)	WORD

Tabela 1. Prefixele cele mai frecvent utilizate în programarea Windows

gram, mesajele structurează procesul de derulare.

Mesajele pot fi de diferite naturi. Ele pot arăta că a avut loc un eveniment de mouse (deplasare sau clic), că s-a apăsat "tasta" sau că fereastra a fost modificată.

Nu trebuie să ne închipuim că un mesaj ar fi ceva deosebit. În principiu el nu este altceva decît un număr fără semn pe 16 biţi. Bineînţeles că în programarea Windows se folosesc nişte prescurtări mai pe înţeles decît acele cifre seci. Toate mesajele încep cu "WM" (Windows Message).

Prelucrarea evenimentelor

O aplicaţie scrisă în C, care utilizează SDK-ul, trebuie să conţină apelul funcţiei "WinMain". Această rutină reprezintă de fapt punctul de intrare în programul propriu-zis. Declaraţia arată astfel:

int PASCAL WinMain (HANDLE hInstance, HANDLE hPrevInstance, LPSTR lpszCmdLine, int cmdShow)

"WinMain" trece prin două faze. Prima este aşa-numita iniţializare de program, cea de-a doua poartă numele de buclă de mesaje. Complexitatea lui "WinMain" constă în iniţializare, care constă la rîndul ei în executarea a trei rutine de bibliotecă Windows şi anume: "RegisterClass", "Create Window" şi "ShowWindow".

Prin parametrul "hInstance" programului i se comunică numele său. Acest nume este necesar identificării. Unele rutine de bibliotecă Windows necesită acest Handle.

Windows caută numele modulului ascuns în fişierul executabil. Acesta poate fi adus într-un format inteligibil utilizatorului cu ajutorul utilitarului "EXEHDR". Dacă se găseşte numele unui modul deja executat, atunci se porneşte o altă instanţă program. Prin parametrul "hPrevInstance" programul află că mai are o "rudă". Programele înrudite folosesc acelaşi cod de program şi aceleaşi resurse (nu şi în mod obligatoriu), în schimb datele diferă între ele.

Linia de comandă "lpszCmdLine" conţine convenţia de apel a programului.

Ultimul parametru a lui "WinMain" este "cmdShow". Aceasta informează programele asupra următoarei operaţii ce va avea loc asupra ferestrei principale.

Bucla de mesaje dă posibilitatea unei interogări continue a mesajelor. Trebuie avută în vedere mereu menţinerea comunicaţiei între părţile individuale ale programului şi alte aplicaţii deschise.

O aplicaţie minimă Windows constă dintr-o fereastră, un meniu sistem, butoanele de minimizare şi de maximizare precum şi din domeniul Client. Toate caracteristicile (deplasare, modificarea mărimilor) sînt preluate de către rutinele bibliotecii Windows. Utilizatorul poate stabili cu ajutorul WRT forma cursorului şi simbolul pentru aplicaţie, pe care le poate integra ulterior.

Listingul 1.1 prezintă o aplicaţie Windows minimală, în care FIRST.C reprezintă programul propriu-zis, în timp ce FIRST.RC, FIRST.ICO şi FIRST.CRS conţine resurse care se integrează în timpul creării fişierului EXE. În plus link-editor-ul Microsoft mai necesită încă un fişier de definire a modulelor, care descrie structura şi construcţia unui program. Este foarte avantajoasă generarea unui fişier MAKE. Astfel se poate economisi scrierea anevoioasă a liniilor necesare de comandă în cazul unor compilări repetate. Această minimă structură a aplicaţiei trebuie folosită în toate cazurile în care SDK-ul îşi găseşte utilitate.

Listingul 1.2 rezolvă aceeaşi soluţie minimală. Biblioteca de clase "Object Windows" uşurează procesul creării unui program. Necesară este doar declararea unui urmaş al lui "TApplication". Orice metodă, care depăşeşte volumul minim (al aplicaţiei) trebuie definită din nou.

Partea a II-a va trata în principal prelucrarea mesajelor. Miezul oricărui program Windows constă din cel puţin o funcţie care poate recepţiona şi prelucra mesaje. De fapt tratarea mesajelor determină modul de comportare al programului.

Se va vedea însă, că programatorul va trebui să se ocupe doar de mesajele care sînt relevante pentru aplicaţia sa. Celelalte mesaje pot fi prelucrate de către "Default Windows Procedure", dacă comportamentul standard al ferestrei rămîne neschimbat. În caz contrar, mesajele care trebuie prelucrate diferit, vor trebui captate.

(I.M.)

Obiectele interfeţei utilizator

Fereastra: este o parte a aplicaţiei. Ea foloseşte structurarea celorlalte obiecte şi răspunde de schimbul de mesaje din sistem.

Icon: este un simbol, care caracterizează o parte a ferestrei ca domeniu de acces asupra unei anumite funcţii. Astfel: meniul sistem, icon-ul de minimizare şi de maximizare sînt domenii de pe ecran, pe care un clic de mouse provoacă o anumită reacţie.

Meniuri: Windows oferă cinci tipuri de meniuri: de sistem, de bare, tip Pull-down, imbricate şi Tear-off. Ele conţin comenzi şi opţiuni de program. Cel mai simplu este meniul sistem, deoarece Windows se îngrijeşte ca funcţionarea comenzilor să fie unitară în întregul sistem.

Cursor: Cursorul este un bitmap şi se deplasează pe ecran corespunzător cu deplasarea unui mouse sau a unei alte unităţi de indicare pe ecran. Aspectul cursorului poate fi modificat (săgeată, stop-hand, clepsidră sau altele). Astfel cursorul poate fi adaptat cerinţelor aplicaţiei şi uşurează perceperea optică a detaliilor unei aplicaţii (de ex.: clepsidra la transferul datelor pe harddisk).

Caret: este denumirea cursorului tastaturii, deoarece prin cursor a fost deja numit arătătorul mouse-ului. Fereastra căreia îi este arondată tastatura poate folosi un Caret pentru a arăta că sînt posibile intrări de date prin intermediul tastaturii. Interfeţele utilizator Windows nu permit decît folosirea unui singur caret. Înseamnă că atunci cînd o fereastră primeşte controlul tastaturii, ea trebuie să producă un caret, iar atunci cînd ea pierde controlul, va trebui să-l distrugă.

Scrollbar: Nu întotdeauna fereastra poate cuprinde întregul domeniul al obiectelor de date. Afişarea barelor de defilare exemplifică această stare de fapt. Cu ajutorul scrollbar-elor utilizatorul poate modifica domeniul de afişare al obiectului de date.

Boxe de dialog: Aceste boxe reprezintă posibilităţi standardizate de introducere date de către utilizator. Ele sînt necesare, de exemplu, atunci cînd după lansarea unei comenzi mai sînt necesare informaţii suplimentare pentru executarea ei corectă. Acestea pot fi cerute şi respectiv corectate prin intermediul unei boxe de dialog.

Instalarea sistemului de operare Netware 2.2

Conceperea şi instalarea reţelelor sub Novell Netware ca şi actualizarea de la Netware 286 la 386 stau în centrul acestei noi serii de articole. În prima parte ne vom ocupa de instalarea lui Netware 2.2.

Netware 2.2 (denumit adesea şi "Netware 286") este urmaşul lui Netware 2.1. Versiunea 2.2 înlocuieşte toate versiunile anterioare 2.1x şi este disponibilă pentru configuraţii diferite. Ea este oferită pentru 5, 10, 50 şi 100 de utilizatori. În ceea ce priveşte numărul maxim de utilizatori care pot lucra simultan nu este însă nici o diferenţă între diferitele configuraţii. Procedurile diferite de instalare dispar.

Caracteristicile de putere ale lui Netware 2.2 le cuprind pe cele conţinute pînă acum de SFT Netware şi includ şi drepturi de acces şi utilizare din Netware 3.11. Novell s-a adaptat astfel evoluţiilor de pe piaţă şi a simplificat mai ales procedura de instalare. Cu versiunea 2.2, Netware a ajuns la a 9-a generaţie şi Novell şi-a ţinut astfel cuvîntul, atunci cînd a promis ca pe lîngă versiunea 3 să dezvolte în continuare şi Netware versiunea 2.

Netware, a noua

Nu întotdeauna trebuie utilizat Netware 3.11. Mai ales în cazul reţelelor mici şi atunci cînd sînteţi la prima experienţă privind lucrul în reţea se recomandă să utilizaţi Netware 2.2 deoarece cerinţele şi deci şi costurile nu sînt tot atît de ridicate.

Cîteva observaţii preliminare - pentru instalarea lui Netware 2.2 este necesar un calculator cu statut special pe care Netware 2.2 să ruleze ca sistem de operare şi care să preia controlul şi comanda reţelei. Acest calculator va fi desemnat ca "Fileserver".

Fileserver şi Workstation

Fileserver-ul lucrează aşadar sub un sistem de operare special care se cheamă "Netware" şi care nu se bazează pe sistemul de operare DOS. Din această cauză Netware nu este supus limitărilor S.O. DOS (de exemplu limitarea la maxim 640 KByte RAM care să poată fi adresaţi direct).

Celelalte calculatoare care vor funcţiona ca posturi de lucru în reţea vor fi desemnate ca "Workstations". În fiecare staţie de lucru, ca şi în fileserver, se va introduce cîte o cartelă de reţea, şi aceste cartele vor fi legate între ele prin cablu. În ceea ce priveşte cablajul nu vom intra în detalii în acest moment. Deocamdată este suficientă o construcţie schematică (fig. 1).

Abia după ce legătura fizică între calculatoare este realizată este posibilă comunicaţia. Aceste legături presupun anumite limitări, de exemplu încăperile unei secţii, sau mai tîrziu clădirea firmei. De aceea se vorbeşte despre o reţea "LAN" (Local Area Network), deci despre o reţea care se limitează la un domeniu local.

Într-o reţea resursele (datele programelor sau fişierele aplicaţiei, imprimantele, ş.a.m.d.) pot fi partajate între utilizatori. Pentru aceasta este necesară o anumită logică de comandă. Pe lîngă staţiile de lucru sînt necesare deci şi calculatoare care să preia anumite sarcini de comandă ("Server"). Unul dintre acestea a fost deja amintit - fileserver-ul. Acest calculator reprezintă nucleul reţelei. Resursele care sînt cuplate la fileserver stau, mai mult sau mai puţin, la dispoziţia tuturor utilizatorilor. Acestea sînt în special harddisk-urile şi imprimantele instalate ca atare în reţea.

Fileserver-ul trebuie să îndeplinească anumite cerinţe: el ar trebui să fie cel puţin un calculator compatibil 100% 286. Procesoarele din generaţii mai noi sînt mai indicate.

Fileserver-ul - un calculator nobil

Un calculator AT cu tactul de 6 MHz nu satisface necesităţile de viteză ale unei reţele. Este mai indicat un AT cu tactul de 16 MHz sau şi mai bine un 386.

Acest calculator trebuie să aibă memoria de cel puţin 1,5 MByte RAM, se recomandă 2,5 MByte. Adesea memoria unui calculator poate fi crescută doar în paşi de MByte întregi, de aceea este bine dacă fileserver-ul dispune de 4 MByte RAM.

Dacă în reţea se instalează un harddisk cu o capacitate mai mare de 70 MByte, pentru fiecare 100 MByte suplimentari fileserver-ul ar trebui să aibă cîte un Mbyte RAM în plus. Dacă pe server se vor instala şi alte aplicaţii sub forma unor aşa-numite "VAP" (Value Added Processes) va trebui să se ţină seama şi de necesarul de memorie al acestora.

Pe fileserver Netware poate fi instalat în două moduri: în mod "Dedicated", caz în care server-ul stă în exclusivitate la dispoziţia sistemului de operare Netware; în modul "Non-Dedicated" pe lîngă Netware mai rămîne disponibil spaţiu şi pentru un proces DOS, astfel încît calculatorul poate fi utilizat şi ca Workstation. Din cauza măsurilor de siguranţă care trebuie luate, modul Non-Dedicated "înghite" încă un MByte de memorie.

Dedicat sau nededicat, aceasta e întrebarea

Acum urmează să se instaleze harddisk-ul şi cartela de reţea în Fileserver.

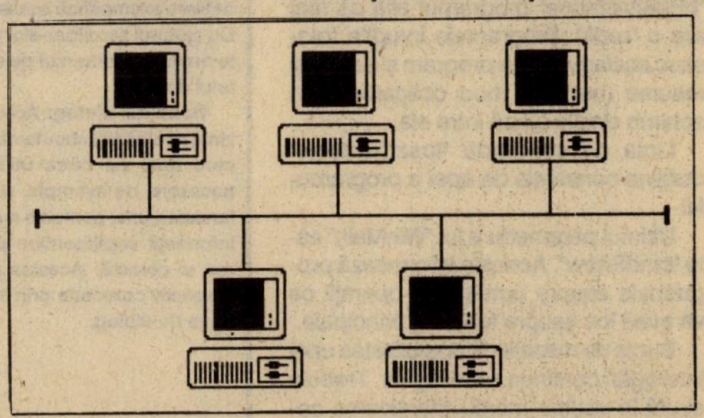


Fig. 1. Construcţia schematică a unei reţele

După ce instalarea hardware-ului este încheiată se poate începe instalarea sistemului de operare Netware.

Porniţi copierea! Nu, nu a dischetelor. Înainte de a face copii de siguranţă după dischetele originale citiţi capitolul 1 din cartea "Installing/Maintaining the Network". Acolo veţi găsi mai multe pagini cu "Worksheets", care trebuie copiate. Important: în Worksheets-urile cu pricina trebuie scrise valorile setate, pentru a avea mai tîrziu o documentaţie a instalării şi a vă putea lămuri rapid problemele care ar putea să apară.

După etapa Worksheets puteţi începe copierea dischetelor. În pachet, înafară de Netware-Box veţi găsi un alt pacheţel cu inscripţia "SERIALIZED DISKETTE", care conţine o dischetă cu numele "OSOBJ". Aceasta stabileşte configuraţia - deci dacă în reţea se vor putea cupla 5, 10, 50 sau 100 de utilizatori. În plus aceasta conţine numărul de serie Netware.

Comparativ cu Netware 2.1x numărul de dischete a scăzut simţitor. Acest lucru s-a realizat prin livrarea produsului doar pe dischete High-Density (1,2 respectiv 1,44 MByte). În varianta 5 1/4 inch, furnitura conţine 18 dischete.

Nu toate aceste dischete trebuie copiate. Chiar mai mult, este suficient să se copieze dischetele cu inscripţiile "SYSTEM-1", "SYSTEM-2", "OSEXE" şi "OSOBJ" (utilizaţi comanda DISKCOPY).

În continuare va trebui să luaţi o altă hotărîre care va influenţa în mod hotărîtor durata instalării. Instalarea poate decurge în întregime pe fileserver. Pentru aceasta se va porni instalarea de pe discheta "SISTEM-1", lansînd în execuţie programul "INSTALL". Dischetele care vor fi necesare în continuare vor fi cerute de programul de instalare. Cu această metodă ajungi curînd să te simţi ca un diskjockey.

Dacă dispuneţi deja de o reţea cu un Fileserver Netware şi nu doriţi un upgrade (o actualizare), atunci există o metodă mai elegantă şi care necesită mai puţin timp. Pentru aceasta este necesară o staţie de lucru care să dispună de un lector de dischete capabil să citească dischetele din furnitura Netware 2.2. Această staţie de lucru poate fi chiar viitorul fileserver, care este "agăţat" în reţeaua existentă ca post de lucru, dacă acest lucru este posibil (de exemplu prin înlocuirea unei staţii de lucru) şi care dispune de shell-ul Netware pentru viitorul Fileserver (mai ales de fişierul IPX.COM).

În acest caz trebuie să deschideţi o sesiune de lucru pe fileserver-ul disponibil (să vă logaţi?!), dar trebuie să aveţi grijă să aveţi drepturi de scriere într-un anumit director. Acest lucru poate fi verificat cu ajutorul comenzii "RIGHTS" (RIGHTS director). În continuare aveţi nevoie de cca. 18 MByte spaţiu liber pe un harddisk al fileserver-ului. Pentru a verifica acest criteriu puteţi utiliza una dintre comenzile "CHKVOL" sau "VOLINFO". Urmează să realizaţi un Drive Mapping pe volumul, corespunzător, de ex. cu comanda: "MAP N:=SYS:". Şi pe această unitate, deci pe N, trebuie apoi să treceţi.

Ca alternativă la utilizarea unei staţii de lucru se poate utiliza şi un calculator DOS cu un spaţiu de memorare liber de cel puţin 18 MByte. Lucrul pe harddisk are avantajul că modificările se pot face foarte repede, fără a mai trebui să se introducă discheta potrivită.

Odată ce vă aflaţi pe o unitate cu capacitate de memorare suficientă va trebui să creaţi un nou director "NETWARE" şi să vă mutaţi în acesta. Pe discheta "SYSTEM-1" se găseşte un program cu numele "UPLOAD" care poate fi apelat cu comanda "A:UPLOAD". La "destination path" va trebui să specificaţi direcorul NETWARE tocmai creat, specificînd şi unitatea. Apoi vi se va cere să introduceţi dischetele Netware într-o anumită succesiune.

Upload creează în directorul "NETWARE" o structură de directoare care corespunde dischetelor individuale. Apoi poate începe instalarea propriuzisă. Din directorul "NETWARE" se va apela programul "INSTALL".

Aşa cum se poate vedea din fig. 2, instalarea cu programul INSTALL constă din patru module diferite. Meniul principal oferă opţiunile "Basic Installation", "Advanced Installation", "Maintain existing System", respectiv "Upgrade from Netware v 2.x". Primele două opţiuni se ocupă cu instalarea simplă sau avansată, a treia permite modificarea unui server deja existent, ultima opţiune serveşte actualizării (upgrade) unei versiuni deja existente, dar mai vechi, pe fileserver la versiunea Netware 2.2.

Opţiunea "Basic Installation" impune instalarea cu parametrii de bază. Opţiunea "Advanced" permite modificarea oricărui parametru. Ecranul cu mesajul de bun venit poate fi părăsit tastînd <ENTER>. În acest moment pe moni-

Nr. modul	Nume modul	Executabil pe	Parametru pentru INSTALL
1	Operating System Generation	viitorul fileserver, harddisk-ul altui calc.	Basic: niciunul Adv. : -E
2	Linking and Configuring	viitorul fileserver, harddisk-ul altui calc.	niciunul
3	ZTEST (Track Zero Test)	viitorul fileserver	-L
4	File Server Definition	viitorul fileserver	-F

Fig. 2. Parametrii de apel pentru "INSTALL"

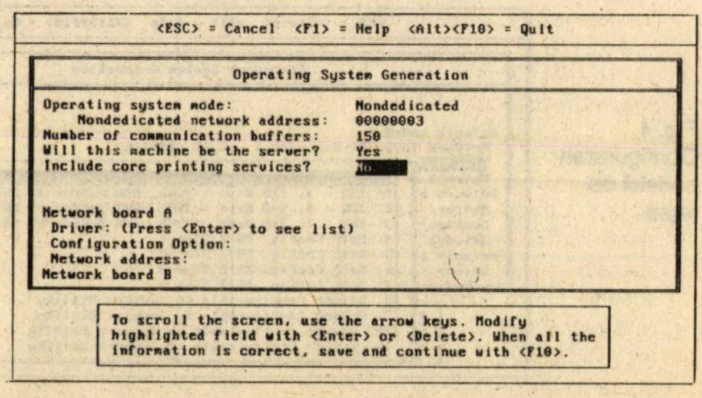


Fig. 3. Modulul "Operating System Generation"

tor va apare primul modul al instalării: "Operating System Generation: (asemănător cu fig. 3).

Primul cîmp determină modul de lucru al fileserver-ului. "Dedicated" va avea ca efect un server dedicat pe care nu va putea rula decît Netware. "Non-Dedicated" permite simultan şi un proces DOS, astfel încît server-ul poate fi utilizat şi ca staţie de lucru.

Avantajul celei de-a doua variante, economisirea costurilor pentru o staţie de lucru implică însă şi dezavantaje semnificative: pe de o parte pe Fileserver nu pot fi rulate programe care utilizează memoria extinsă (de exemplu Windows 3.0). Pe de altă parte un program care rulează ca proces DOS se poate bloca şi poate determina astfel căderea server-ului şi deci a întregii reţele.

Procesul DOS al unui server nededicat necesită o adresă în reţea, care este văzută ca o adresă logică de cartelă de reţea. Această "Nondedicated Network Adress" trebuie să fie diferită deci de toate adresele cartelelor de reţea instalate!

Operation Buffer

Numărul buffer-elor de comunicaţie (Communication Buffer) specifică numărul blocurilor rezervate pentru buffer-e pe fileserver pentru interogări din partea staţiilor de lucru. Conţinutul buffer-elor este prelucrat de acesta cu urgenţa maximă. Un bloc ocupă 512 Byte de memorie. Dacă buffer-ul este plin şi nu mai poate prelua alte interogări, staţiile de lucru vor trebui să-şi repete cererea.

O valoare prea mică influenţează deci negativ viteza staţiilor de lucru, o valoare prea mare determină o supraîncărcare inutilă a memoriei fileserver-ului. Se pot da valori cuprinse între 40 şi 1000. Valoarea standard de 150 de buffer-e este o bună valoare de pornire.

Dependent de faptul dacă instalarea s-a lansat de pe harddisk sau nu, va trebui răspuns la întrebarea : "Will this machine be the server?". Dacă se răspunde cu "No" vor fi rulate numai primele două module de instalare. Apoi dischetele vor trebui trecute printr-o operaţie "download" şi cu aceste dischete procesul de instalare va continua, pornind de la al treilea modul, pe viitorul fileserver. Dacă instalarea se face de pe harddisk-ul unui calculator DOS sau de pe o staţie de lucru va trebui să răspundeţi "No".

"Yes", dimpotrivă, determină continuarea instalării, cu urmarea că după parcurgerea celui de-al treilea modul veţi pierde toate datele care le aveaţi pe harddisk-ul fileserver-ului! Dacă instalarea se execută de pe un server existent - viitorul fileserver fiind momentan legat ca staţie de lucru - va trebui să răspundeţi "Yes".

La "Include Core Printing" se specifică dacă tratarea imprimantelor trebuie integrată în reţea ("Yes"). Acest răspuns are ca efect posibilitatea cuplării imprimantelor din reţea la fileserver (max. 5 buc.). Dacă în reţea ar trebui integrate şi imprimante cuplate la staţii de lucru, noţiunea Netware "Printservers" vorbeşte de la sine şi deci aici nu poate fi luat în considerare decît un răspuns "No". Despre Printserver vom vorbi mai tîrziu.

Urmează să stabiliţi specificaţiile pentru cartela de reţea (Network board) şi harddisk-ul (Channel) fileserver-ului. În acesta pot fi instalate pînă la 4 cartele de reţea, care vor fi denumite de la "LAN A" pînă la "LAN D". LAN A, respectiv "Networkboard A" este obligatorie, toate celelalte sînt opţionale. Un sfat: procesul DOS al unui server nededicat îl puteţi denumi "LAN E".

Tipul cartelei de reţea îl puteţi specifica în cadrul opţiunii "Driver", după ce aţi tastat <ENTER>. Va trebui să selectaţi acel driver care corespunde cartelei de reţea disponibile. Dacă utilizaţi o cartelă Ethernet, acestea sînt adesea compatibile cu cartelele Novell NE-2000, al căror driver-e există deja în furnitura Netware. Cartelele Arcnet pot utiliza adesea driver-ul "Netware-RX-Net" care este de asemenea în furnitură.

Dacă aveţi dubii va trebui să parcurgeţi manualul cartelei de reţea (majoritatea suportă Netware). Dacă nu găsiţi nici un driver potrivit, atunci e de sperat că odată cu cartela de reţea vi s-a livrat şi o dischetă cu un driver pentru Netware. Adesea pe aceste dischete există doar drivere pentru Netware 2.1x. Aceste driver-e pot fi utilizate totuşi şi pentru Netware 2.2. După ce aţi tastat <Insert> şi aţi introdus discheta cu pricina puteţi selecta driver-ul suplimentar.

Cartelele de reţea utilizează configuraţii diferite pentru întreruperi. Configuraţia instalată trebuie specificată în cadrul opţiunii "Configuration Option" (fig. 4).

Opţiunea 0 corespunde de regulă configuraţiei de lucru. Valorile standard pentru Ethernet sînt Interrupt 3 şi I/0 Adresse 300, pentru Arcnet Interrupt 2, I/0 - Adresse 2EO respectiv DOOO ca buffer RAM. Aceste valori depind desigur de hardware, ca de pildă de interfeţele seriale şi paralele, cartela grafică, etc. Este indicat să verificaţi datele pentru cartela de reţea.

Fiecare cartelă de reţea necesită o adresă de reţea neambiguă (Network Address). Adresa de reţea este un număr hexazecimal cu opt poziţii (0 pînă la F).

Această procedură (Driver, Configuration Option, Network Address) trebuie repetată pentru toate celelalte cartele de reţea. Toate valorile date pînă în acest punct va trebui să le copiaţi întocmai în copiile worksheet.

(continuarea în numărul următor)

(R.M.)

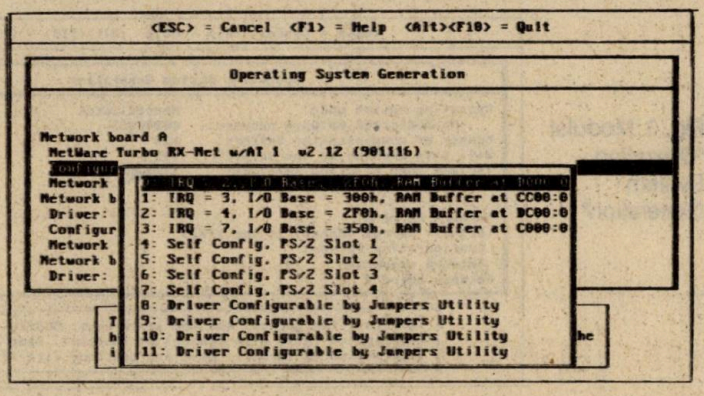


Fig. 4. Configurarea cartelei de reţea

str. Gheorghe Doja 36
Tîrgu-Mureş ROMÂNIA
Adresa poştală: C.P. 64, O.P. 1
4300 Tîrgu-Mureş
ROMÂNIA
Telefon/Telefax: 004-954-31660

AUTHORIZED REMARKETER
- IBM PERSONAL COMPUTER PRODUCTS

AUTHORIZED REMARKETER
- IBM PERSONAL SYSTEM/1 COMPUTER

vă oferă

CALCULATOARE PERSONALE

din gama PS/1 şi PS/2

Raportul preţ/performanţă este în favoarea Dvs. !

Cîte ceva despre VGA

Relativ puţine informaţii sînt de găsit în ce priveşte programarea cartelelor grafice VGA - nu în ultimul rînd datorită lipsei unui standard acceptat.

Organizarea memoriei grafice la VGA.

Memoria grafică este organizată pe patru plane, fiecare plan avînd capacitatea de 65536 octeţi (64 Ko). Planele sînt numerotate de la 0 la 3, şi sînt adresate paralel, astfel că este necesar un spaţiu de adresare de doar 64 Ko (de la A000:0000 la A000:FFFF).

Cum se adresează memoria ecran?

a) Scrierea. Porturile "Sequence Controller Index" se programează astfel:

-scriere in memoria ecran:

```
Port[$3C4]:=2; (Map Mask Register)
```

- precizarea planului:

```
Port[$3C5]:=valoare;
```

(cuprinsă între 1 si 15)

Bitul i (i intre 0 si 3) al valorii înscrise la portul $3C5 are următoarea semnificaţie: informaţia va fi înscrisă în planul i dacî bitul i are valoarea 1. Pot fi puşi pe 1 unul sau mai mulţi biţi; în acest al doilea caz informaţia va fi înscrisă simultan în mai multe plane. Exemplu:

```
Port[$3C4]:=2;
Port[$3C5]:=5; Mem[$A000:5]:=201;
```

Valoarea 201 va fi inscrisă simultan în planele 0 si 2, la adresa relativă 5.

b) Citirea. Porturile "Graphics Controller Index" se programează astfel:

- citire din memoria ecran:

```
Port[$3CE]:=4; (Read Map Register)
```

- precizarea planului:

```
Port[$3CF]:=valoare;
```

(cuprinsă între 0 si 3)

Valoarea înscrisă la portul $3CF dă numărul planului de unde se va citi informaţia. Evident, la un moment dat putem citi dintr-un singur plan.

Exemplu:

```
Port[$3CE]:=4;
Port[$3CF]:=1; B:=Mem[$A000:9];
```

Variabila B (de tip byte) va primi valoarea aflată în planul 1 la adresa relativă 9.

Programarea paletei de culori pentru modurile cu 256 culori.

Cartela VGA poate afişa simultan 256 culori dintr-o paletă de 262144 (2^{18}) culori.

Pentru programarea paletei de culori se apelează întreruperea BIOS $10 cu următoarele valori în regiştri:

```
ah = $10; al = $10; bh = 0;
bl = index-culoare;
ch = i-green; cl = i-blue; dh = i-red;
```

Semnificaţia valorilor din regiştri:

index-culoare = valoare între 0 si 255;

i-red, i-green, i-blue = intensitatea corespunzătoare componentelor Roşu, Verde şi Albastru;

Modul 320x240x256

În acest mod ecranul se prezintă astfel:

```
P( 0, 0)  P( 1, 0)  ...  P(319, 0);
P( 0, 1)  P( 1, 1)  ...  P(319, 1)
  ...       ...     ...    ...
P(0,239)  P(1,239)  ...  P(319,239)
```

Cele patru plane de memorie sînt organizate astfel:

```
Planul 0:
P( 0, 0)  P( 4, 0)  ...  P(316, 0);
P( 0, 1)  P( 4, 1)  ...  P(316, 1)
  ...       ...     ...    ...
P(0,239)  P( 4,239) ...  P(316,239)
Planul 1:
P( 1, 0)  P( 5, 0)  ...  P(317, 0);
P( 1, 1)  P( 5, 1)  ...  P(317, 1)
  ...       ...     ...    ...
P(1,239)  P(5,239)  ...  P(317,239)
Planul 2:
P( 2, 0)  P( 6, 0)  ...  P(318, 0);
P( 2, 1)  P( 6, 1)  ...  P(318, 1)
  ...       ...     ...    ...
P(2,239)  P(6,239)  ...  P(318,239)
Planul 3:
P( 3, 0)  P( 7, 0)  ...  P(319, 0);
P( 3, 1)  P( 7, 1)  ...  P(319, 1)
  ...       ...     ...    ...
P(3,239)  P(7,239)  ...  P(319,239)
```

Am indicat între paranteze coordonatele x şi y ale pixelilor.

Programarea modului 320x240x256 se realizează astfel:

- se apelează funcţia BIOS pentru programarea în modul standard, 320x200x256;
- se programează cîţiva regiştri ai controlorului VGA pentru a obţine 480 linii pe verticală şi apoi pentru a scana de două ori aceeaşi linie (astfel se obţine o rezoluţie de 240 linii pe verticală); rezoluţia pe orizontală se păstrează (320 pixeli pe linie);
- se programează de asemenea controlorul VGA pentru a folosi toate cele patru plane de memorie, similar cu modul 640x480x16 (deoarece modul 320x200x256 foloseşte un singur plan de memorie).

De ce am ales modul 320x240x256 ?

Acest mod prezintă mai multe avantaje faţă de celelalte moduri:

- fiecărui pixel îi corespunde un octet în memoria ecran, de unde rezultă o viteză de prelucrare de cîteva ori mai mare faţă de modurile cu 16 culori, unde operaţiile la nivel de pixel sînt destul de laborioase;
- acest mod funcţionează pe orice tip de VGA;
- faţă de celelalte moduri cu 256 culori, acest mod prezintă de asemenea avantaje:
- densitatea pixelilor este aceeaşi pe orizontală ca şi pe verticală, deci dacă se desenează un cerc sau un pătrat, de exemplu, nu trebuie făcute operaţii suplimentare de ajustare a coordonatelor x şi y;
- este posibilă paginarea memoriei ecran.

Modul 360x480x256

Acest mod diferă de cel prezentat anterior prin faptul că cele patru plane de memorie ecran sînt organizate fiecare ca o matrice de dimensiuni 480x90. Programarea acestui mod diferă de asemenea prin faptul că se programează atît rezoluţia pe orizontală (360 pixeli pe linie, deci 90x4), cît şi cea pe verticală (480 linii, deci aici fiecare linie va fi scanată o singură dată).

În articolul de faţă sînt prezentate cîteva programe scrise in Turbo Pascal care ilustrează folosirea modului 320x240x256 (Xmod + Xgrafica) şi a modului 360x480x256 (Cmod + Cgrafica). Precizăm că programele au fost testate pe mai multe calculatoare compatibile IBM-PC, cu VGA de tip Oti's şi OAK.

Într-un articol viitor vom prezenta un program de animaţie folosind tehnicile de paginare a memoriei ecran. Programul va folosi modul 320x240x256 datorită avantajelor pe care le prezintă faţă de alte moduri grafice.

mat. Nelu Cozac

```
uses cgrafica,crt;
var i,j:integer;
begin
  csetgraph;
  csetcolor(5,63,0,63);
  csetcolor(9,0,0,63);
  for i:=0 to 479 do
    begin
      cwritepixel(0,i,5);
      cwritepixel(1,i,5);
    end;
  for i:=0 to 479 do
    begin
      cwritepixel(359,i,5);
      cwritepixel(358,i,5);
    end;
  for j:=0 to 359 do
    begin
      cwritepixel(j,0,5);
      cwritepixel(j,1,5);
    end;
  for j:=0 to 359 do
    begin
      cwritepixel(j,479,5);
      cwritepixel(j,478,5);
    end;
  for i:=2 to 477 do
    for j:=2 to 357 do
      cwritepixel(j,i,9);
  delay(2000);
  port[$3c4]:=2; port[$3c5]:=1; delay(20);
  fillchar(mem[$a000:0],480*90,#0);
  port[$3c4]:=2; port[$3c5]:=2; delay(20);
  fillchar(mem[$a000:0],480*90,#0);
  port[$3c4]:=2; port[$3c5]:=4; delay(20);
  fillchar(mem[$a000:0],480*90,#0);
  port[$3c4]:=2; port[$3c5]:=8; delay(20);
  fillchar(mem[$a000:0],480*90,#0);
  cterminate;
end.
```

CMOD.PAS

```
uses xgrafica,crt;
var i,j:integer;
begin
  xsetgraph;
  xsetcolor(5,63,0,63);
  xsetcolor(9,0,0,63);
  for i:=0 to 239 do
    begin
      xwritepixel(0,i,5);
      xwritepixel(1,i,5);
    end;
  for i:=0 to 239 do
    begin
      xwritepixel(319,i,5);
      xwritepixel(318,i,5);
    end;
  for j:=0 to 319 do
    begin
      xwritepixel(j,0,5);
      xwritepixel(j,1,5);
    end;
  for j:=0 to 319 do
    begin
      xwritepixel(j,239,5);
      xwritepixel(j,238,5);
    end;
  for i:=2 to 237 do
    for j:=2 to 317 do
      xwritepixel(j,i,9);
  delay(2000);
  xsetcolor(5,0,63,0);
  xsetcolor(9,63,63,0);
  delay(2000);
  xterminate;
end.
```

XMOD.PAS

```
unit xgrafica; {$D-,L-,S-}

interface uses dos;

procedure xsetgraph;
procedure xwritepixel(x,y:word; p:byte);
function xreadpixel(x,y:word):byte;
procedure xsetcolor(c,r,g,b:byte);
procedure xterminate;

implementation

var regs:registers; i,j,k:word;
  me:array[0..239,0..79] of byte absolute $a000:0;

const valori:array[1..10] of word=
  ($0d06,$3e07,$4109,$ea10,$ac11,
   $df12,$0014,$e715,$0616,$e317);

procedure xsetgraph;
begin
  regs.ax:=19; intr(16,regs);
  portw[$3c4]:=$604;
  portw[$3c4]:=$100;
  port[$3c2]:=$e3;
  portw[$3c4]:=$300;
  port[$3d4]:=$11;
  port[$3d5]:=port[$3d5] and $7f;
  for i:=1 to 10 do portw[$3d4]:=valori[i];
  portw[$3c4]:=$f02;
  fillchar(me,240*80,#0);
end;

procedure calcule(x,y:word);
begin
  k:=x and 3; i:=y; j:=x div 4;
end;

procedure xwritepixel(x,y:word; p:byte);
begin
  calcule(x,y);
  port[$3c4]:=2; port[$3c5]:=1 shl k;
  me[i,j]:=p;
end;

function xreadpixel(x,y:word):byte;
begin
  calcule(x,y);
  port[$3ce]:=4; port[$3c5]:=k;
  xreadpixel:=me[i,j];
end;

procedure xsetcolor(c,r,g,b:byte);
begin
  with regs do
    begin
      ah:=16; al:=16; bh:=0; bl:=c;
      ch:=g; cl:=b; dh:=r;
    end;
  intr(16,regs);
end;

procedure xterminate;
begin
  regs.ax:=3; intr(16,regs);
end;

end.
```

XGRAFICA.PAS

```
unit cgrafica; {$D-,L-,S-}

interface uses dos;

procedure csetgraph;
procedure cwritepixel(x,y:word; p:byte);
function creadpixel(x,y:word):byte;
procedure csetcolor(c,r,g,b:byte);
procedure cterminate;

implementation

var regs:registers; i,j,k:word;
  me:array[0..479,0..89] of byte absolute $a000:0;

const valori:array[0..9] of byte=
  ($6d,$59,$5a,$8e,$62,$8b,$0b,$3e,0,$40);

const valorj:array[0..7] of byte=
  ($ea,$8c,$df,$2d,0,$e7,$04,$c3);

procedure csetgraph;
begin
  regs.ax:=19; intr(16,regs);
  portw[$3c4]:=$604;
  portw[$3c4]:=$100;
  port[$3c2]:=$c7;
  portw[$3c4]:=$300;
  port[$3d4]:=$11;
  port[$3d5]:=port[$3d5] and $7f;
  for i:=0 to 9 do
    begin
      port[$3d4]:=j; port[$3d5]:=valori[i];
    end;
  for i:=0 to 7 do
    begin
      port[$3d4]:=i+16; port[$3d5]:=valorj[i];
    end;
  portw[$3c4]:=$f02;
  fillchar(me,480*90,#0);
end;

procedure calcule(x,y:word);
begin
  k:=x and 3; i:=y; j:=x div 4;
end;

procedure cwritepixel(x,y:word; p:byte);
begin
  calcule(x,y);
  port[$3c4]:=2; port[$3c5]:=1 shl k;
  me[i,j]:=p;
end;

function creadpixel(x,y:word):byte;
begin
  calcule(x,y);
  port[$3ce]:=4; port[$3c5]:=k;
  creadpixel:=me[i,j];
end;

procedure csetcolor(c,r,g,b:byte);
begin
  with regs do
    begin
      ah:=16; al:=16; bh:=0; bl:=c;
      ch:=g; cl:=b; dh:=r;
    end;
  intr(16,regs);
end;

procedure cterminate;
begin
  regs.ax:=3; intr(16,regs);
end;

end.
```

CGRAFICA.PAS

Cine indexează mai bine ?

Testul	486 /25MHz/16Mb 200Mb/16ms	286/15 MHz/1Mb 40Mb /16ms	286/12MHz /1Mb 40Mb /28ms
Creare si indexare	7'55"	23'28"	32'20"
Interogare 1	32"	55"	1'37"
Interogare 2	9"	20"	30"

```
/*
* CLIPPER 5.0/5.01
*
* DEMCLIP creeaza un fis. secv.indexat in acces
* direct de 100000 articole, apoi interogheaza in
* acces direct 1000 de articole in doua modalitati
*
* compilare:  > clipper demclip /m/n/w/a
* linkeditare: > rtlink fi demclip
*/

STATIC i, j, l := 0
STATIC rest
STATIC secvsf
STATIC Wch_Cit1 [1000]

STATIC m_ab1
STATIC m_ab2
STATIC m_altc1 := "INF.GR1"
STATIC m_altc2
STATIC m_altc3 := "INF.GR2"
STATIC m_altc4

FIELD AB1   // N, 6
FIELD AB2   // C, 2
FIELD ALTC1  // C, 6
FIELD ALTC2  // N, 6
FIELD ALTC3  // C, 6
FIELD ALTC4  // N, 6

PROCEDURE main()

USE DEMCLIP
ZAP
INDEX ON STR(AB1,6)+AB2 TO DEMCLIP
?"     P R O G R A M   D E M C L I P"

? "INDEXARE INCEPUTA:    "+ TIME()
j    := 0
secvsf := 100000
FOR i := 1 TO 50000
   Gen_Fis()
NEXT
? "INDEXARE TERMINATA:   "+ TIME()

USE DEMCLIP READONLY
SET INDEX TO DEMCLIP

? "INTEROGARE 1 INCEPUTA: "+ TIME()
j := 1001
FOR i := 1 TO 500
   Cit1_Fis()
NEXT
? "INTEROGARE 1 TERMINATA: "+ TIME()

? "INTEROGARE 2 INCEPUTA: "+ TIME()
FOR i := 1 TO 1000
   Cit2_Fis()
NEXT
? "INTEROGARE 2 TERMINATA: "+ TIME()
USE

QUIT

// ----------------------------------------

PROCEDURE Gen_Fis

secvsf--

rest := i % 26
Gen_Ch()
m_ab1 := m_altc2 := m_altc4 := i
Scr_Fis()

rest := secvsf % 26
Gen_Ch()
m_ab1 := m_altc2 := m_altc4 := secvsf
Scr_Fis()

RETURN

// ----------------------------------------
PROCEDURE Gen_Ch

LOCAL schimb
STATIC Wch_G2 := { "AB", "CD", "EF", "GH", "IJ",;
                   "KL", "MN", "OP", "QR", "ST",;
                   "UV", "WX", "YZ", "ab", "cd",;
                   "ef", "gh", "ij", "kl", "mn",;
                   "op", "qr", "st", "uv", "wx",;
                   "yz", " " }
rest++
m_ab2     := Wch_G2 [rest]

schimb    := Wch_G2 [rest]
Wch_G2 [rest] := Wch_G2 [27]
Wch_G2 [27]  := schimb

RETURN

// ----------------------------------------
PROCEDURE Scr_Fis

j++
IF j > 95
   j := 0
   l++
   IF l < 1001
      Wch_Cit1 [l] := STR(m_ab1,6)+m_ab2
   ENDIF
ENDIF

SEEK STR(m_ab1,6)+m_ab2
IF FOUND ()
   ? "CHEIE DUBLA: "+STR(m_ab1,6)+m_ab2
ELSE
   APPEND BLANK
   AB1  := m_ab1
   AB2  := m_ab2
   ALTC1 := m_altc1
   ALTC2 := m_altc2
   ALTC3 := m_altc3
   ALTC4 := m_altc4
ENDIF

RETURN

// ----------------------------------------
PROCEDURE Cit1_Fis

SEEK Wch_Cit1 [i]
IF ! FOUND ()
   ? "CHEIE NEGASITA: "+Wch_Cit1 [i]
ENDIF
j--
SEEK Wch_Cit1 [j]
IF ! FOUND ()
   ? "CHEIE NEGASITA: "+Wch_Cit1 [j]
ENDIF

RETURN

// ----------------------------------------
PROCEDURE Cit2_Fis

SEEK Wch_Cit1 [i]
IF ! FOUND ()
   ? "CHEIE NEGASITA: "+Wch_Cit1 [i]
ENDIF

RETURN

// ----------------------------------------
```

DEMCLIP.PRG

Clipper 5.01

Programul a fost compilat cu CLIPPER 5.01 si executat pe 3 calculatoare, cu FILES=50 si BUFFERS=35 in CONFIG.SYS.

Dimens.programului executabil: 223 Ko
Dimens. fişierului de date (dbf): 3,3 Mo
Dimens.fisierului index (ntx): 3,8 Mo
(dupa reindexare: 1,9 Mo)

Observatii:

1. Fişierul creat nu are o dispersie uniformă.
2. Testul 1 este relevant în prelucrări de tip "pe loturi", nu şi în prelucrări conversaţionale, specifice calculatoarelor personale.

În schimb lipsesc teste ca:

- parcurgere secvenţială,
- parcurgere indexată,
- adăugarea, modificarea, ştergerea unui numar de articole cu cod aleator.

3. Fişierul index rezultat după o reindexare se micşorează la jumătate. Proporţiile micşorării se datorează dispersiei neuniforme. Cu un fişier index reorganizat, evident scad şi timpii de parcurgere indexată şi unele interogări. Timpii de interogare din tabel au fost obtinuţi cu fişierul index **ne**reorganizat.

HAMOR Soft SRL
4000 - Sfintu Gheorghe c.p. 140
tel: 923/12139, 15648, fax: 12139

Quattro Pro

Crearea de copii multiple

Dacă într-o tabelă doriţi să copiaţi un domeniu de mai multe ori, în mod normal, trebuie să marcaţi acel domeniu de fiecare dată şi apoi să-l inseraţi în locul dorit. Trucul următor vă ajută să rezolvaţi această problemă mai simplu.

- 1. Selectaţi meniurile *Edit-Names-Create* şi daţi litera "K", de exemplu, ca nume pentru domeniul sursă.
- 2. Marcaţi domeniul care trebuie copiat de mai multe ori şi apoi tastaţi *<Enter>*.
- 3. Mutaţi cursorul în colţul din stînga sus al domeniului în care doriţi să inseraţi prima copie.
- 4. Tastaţi *<Ctrl>+<C>* pentru a apela meniul *Edit-Copy*. În loc să marcaţi din nou domeniul care trebuie copiat, tastaţi *<K>*, după care apăsaţi de două ori tasta *<Enter>*. În acest mod puteţi copia un domeniu de ori cîte ori doriţi.

Salvare fişier la un simplu clic pe mouse

Dacă doriţi să memoraţi un fişier pe disc, apelînd meniurile *File-Save*, şi există deja un fişier pe hard disk cu nume identic, o să primiţi un mesaj de avertizare, care vă va semnala acest lucru. Va trebui să selectaţi *Replace* deşi nu doreaţi altceva decît să faceţi o nouă copie de siguranţă.

Paleta de funcţii selectabile cu ajutorul mouse-ului, aflată în marginea din dreapta a ecranului, vă poate oferi o soluţie mai simplă:

- 1. Selectaţi meniurile *Options-MousePalette*.
- 2. Selectaţi cîmpul paletei mouse căruia doriţi să-i încredinţaţi funcţia de salvare. Pentru aceasta puteţi folosi cea de-a cincea tastă - *5th Button* - care în mod standard poartă inscripţia "5" şi determină apelarea unui macro pentru redarea unui sunet.
- 3. În cadrul submeniului *Text*, daţi o denumire proprie cîmpului, denumire care să nu fie mai lungă de 3 caractere.
- 4. Selectaţi submeniul *Macro* şi tastaţi următoarea succesiune de caractere:

```
/fsr
```

- 5. Dacă selectaţi apoi *Quit*, veţi vedea noul cîmp apărînd în meniul din marginea din dreapta a ecranului. În viitor este suficient să poziţionaţi cursorul mouse-ului pe acest cîmp şi să daţi un clic-mouse.

Revenire rapidă

Quattro Pro dispune de o structură de meniuri ierarhică, cu ajutorul cărora se selectează o anumită funcţie sau se stabilesc anumiţi parametri. De exemplu pentru a selecta o diagramă de tipul 3D, trebuie pătruns pînă pe al treilea nivel de meniuri: *Graph-Graph Type-3D*.

Adesea însă se selectează un astfel de meniu doar pentru a vedea care sînt opţiunile disponibile. Pentru a reveni în tabelă, în acest caz, trebuie apăsată de trei ori tasta *<Esc>*. De aici şi trucul: dacă se tastează *<Ctrl>+<Break>* se sare peste toate nivelele intermediare şi se revine în tabelă.

Cuplare rapidă tabele

Ca şi Lotus 1-2-3 şi Quattro Pro oferă posibilitatea introducerii într-o tabelă a unor aşa-numite formule de cuplare prin intermediul cărora se stabileşte o legătură cu anumite cîmpuri, sau domenii, din alte tabele. Pentru aceasta trebuie făcută o referire externă, specificîndu-se numele tabelei dorite şi al cîpului, sau al domeniului, ca de ex.:

```
+1,14*[F:\qpro\VOT.WQ1]A35
```

Această posibilitate este foarte puternică, dar referinţa implică tastarea multor caractere. Cu ajutorul unui truc, o formulă care să se refere la mai multe tabele se poate scrie mult mai simplu. Să presupunem că există trei fişiere: BRASOV.WQ1, CLUJ.WQ1 şi ARAD.WQ1, care în cîmpul E17 conţin numărul alegătorilor participanţi la vot. Pentru a aduna aceste valori într-o a patra tabelă ar trebui tastată următoarea formulă:

```
@SUM([C:\QPRO\
BRASOV.WQ1]E17;
[C:\QPRO\CLUJ.WQ1]E17;
[C:\QPRO\ARAD.WQ1]E17)
```

În loc să tastaţi o formulă atît de lungă puteţi să rezolvaţi problema mult mai simplu executînd următorii paşi:

- 1. Încărcaţi toate tabelele care trebuie cuplate.
- 2. Treceţi în tabela în care doriţi însumarea.
- 3. Tastaţi formula:

```
@SUM([*]E17)
```

- 4. După ce aţi tastat *<Enter>* această formulă este transformată în formula anterior descrisă. Motivul: asterix-ul va determina cuplarea tuturor fişierelor deschise în acel moment.

Marcare rapidă

De regulă Quattro Pro este compatibil cu Lotus 1-2-3, dar există şi cîteva deosebiri semnificative. Una dintre acestea este de exemplu suprafaţa Pull-Down şi operarea cu mouse. Tocmai la marcarea unui domeniu cu mouse-ul se poate folosi un truc interesant. Atunci cînd se doreşte marcarea unui domeniu mare se poziţionează cursorul în colţul din stînga sus al domeniului, se apasă tasta din stînga a mouse-ului şi ţinîndu-se apăsată se extinde marcajul pînă în colţul din dreapta jos al domeniului.

Mai simplu se poate rezolva problema în modul următor:

- 1. Se poziţionează cursorul în colţul din stînga sus şi se dă un clic pe mouse.
- 2. Fără a se începe marcajul se mută cursorul în colţul din dreapta jos.
- 3. Se apasă tasta din dreapta a mouse-ului şi ţinînd-o apăsată se apasă şi pe tasta din stînga. Quattro Pro extinde deja marcajul pînă la poziţia cu pricina.

Comod este faptul că blocul selectat se poate mări sau micşora, la dorinţă, poziţionînd cursorul într-o nouă poziţie şi apăsînd simultan tastele din dreapta şi din stînga mouse-ului.

Vă rugăm să ne scrieţi părerea dumneavoastră despre această nouă rubrică şi să specificaţi şi pachetele software pentru care aţi dori să prezentăm astfel de trucuri.

Nu uitaţi să ne trimiteţi propriile descoperiri!

Corecţii Borland în unit-ul »editors« (Turbo Vision)

Din păcate, în unit-ul »editors« din TurboVision s-a strecurat o greşeală. Puteţi opera însă şi D-voastră corecturile; trebuiesc schimbate numai cîteva linii în fişierul »editors.pas«, după cum urmează.

Secvenţa

```
LODSW
CMP     BL,DL
```

va trebui să înlocuiască secvenţa

```
LODSW
OR      BL,BL
JE      @@2
CMP     BL,DL
```

din rutina ScanKeyMap. Secvenţa

```
JE      @@4
CMP     BH,DH
JE      @@4
```

înlocuieşte secvenţa

```
    JE      @@4
@2: CMP     BH,DH
    JE      @@4
```

din rutina ScanKeyMap. Secvenţa

```
var
    ShiftState: Byte absolute $40:$17;
    Key: Word;
```

înlocuieşte secvenţa

```
var
    Key: Word;
```

din rutina ConvertEvent; În fine, secvenţa

```
begin
    if (ShiftState and $03 <> 0) and
      (Event.ScanCode >= $47) and
      (Event.ScanCode <= $51) then
      Event.CharCode := #0;
      Key := Event.KeyCode;
```

înlocuieşte secvenţa

```
begin
    Key : = Event.KeyCode;
```

din rutina ConvertEvent. Deoarece dischetele de referinţă erau deja produse iar unit-ul este disponibil în format sursă, Borland s-a decis pentru acest mod de operare a corecţiilor.

Borland

Cod autoreproducător

»Clona.c« este un program care se "autoreproduce". Spre deosebire însă de cei mai mulţi reprezentanţi ai acestei "specii", care datorită unor virulente "oi negre" se bucură de o nemeritată faimă proastă, programul »Clona« nu duplică codul executabil ci produce o imagine exactă a propriei sale surse. (Desigur, pentru acest lucru nu se foloseşte sursa existentă, ca original pentru o copie banală; nu se face nici un fel de referinţă spre un fişier »clona.c« existent.) De fapt, »Clona.c« nu face uz de nici un fel de comenzi explicite pentru ieşire în fişier: duplicarea se obţine prin faptul că output-ul programului este redirectat spre un fişier cu acelaşi nume. Programul se lansează cu sintaxa:

```
clona > clona.c
```

Soluţionarea problemei a început cu căutarea unei instrucţiuni (de ieşire) care să se autoreproducă; mai exact, care să se conţină pe ea însăşi ca argument şi să-l "scoată" pe acesta de două ori. Experimentarea unor rutine de construcţie proprie nu a fost (din fericire) necesară, deoarece rutina »printf« din biblioteca standard C ştie tot ce trebuie pentru acest lucru. Trucul constă în a "împacheta" întregul cod al programului în şirul de formatare şi de a-l folosi suplimentar pe acesta ca argument-şir. Simplificînd, schema de lucru se prezintă de maniera »printf(X,X)«, unde

```
X = "X = %s printf(X,X)"
```

Referinţa înapoi în această structură evită o recursie infinită după modelul:

<instrucţiune-de-ieşire>("<instrucţiune-de-ieşire>")<instrucţiune-de-ieşire>

Ultimul obstacol - ghilimelele care acţionează ca delimitatori de şir şi formatarea textului (fragmentarea lui în linii distincte - se trece folosind constante ASCII. *(vezi şi if 8/91 (mov), pag. 29)*

(G. H./I.F.)

```
1: #include <stdio.h>
2: char a[ ]="#include<stdio.h>%cchar
   a[ ]=%c%s%c;%cvoid main() {printf
   (a,10,34,a,34,10) ;}";
3: void main() {printf (a,10,34,a,34,10) ;}
```

Clona.C

Curs C++ (IV)

(continuare din pag. 48)

```
128public:
129 View::View (int x1, int y1, int x2, int y2, \
130   Lista* p = NULL, Fereastra* e = NULL,\
131   Lista* s = NULL)
132   : Lista (p, e, s),
133 Fereastra (x1,y1,x2,y2)
134 {
135  vizibil = 1;
136 }
137 void View::Afisare()
138 {
139  puttext (get_elem() -> GetX1(),
140   get_elem() -> GetY1(),
141   get_elem() -> GetX2(),
142   get_elem() -> GetY2(),
143   get_elem() -> GetBuf() );
144 }
145};
146
147int main()
148{
149 char tasta;
150 Boolean sfirsit;
151 Fereastra* ff;
152 View* v[5];
153 int z;
154
155 clrscr();
156 ff = new Fereastra (XMin, YMin, XMax, YMax);
157 ff->Umplere ('x',0x16);
158 ff->Afisare();
159
160 v[1] = new View (10, 3, 46, 15);
161 v[1]->Umplere ('A', 0x12);
162
163 v[2] = new View (14, 5, 50, 17);
164 v[2]->Umplere ('B', 0x2f);
165
166 v[3] = new View (18, 7, 54, 19);
167 v[3]->Umplere ('C', 0x40);
168
169 v[4] = new View (22, 9, 58, 21);
170 v[4]->Umplere ('D', 0x38);
171
172 v[5] = new View (26, 11, 62, 23);
173 v[5]->Umplere ('E', 0x51);
174
175 v[1]->set_list (NULL, v[1], v[2]);
176 v[2]->set_list (v[1], v[2], v[3]);
177 v[3]->set_list (v[2], v[3], v[4]);
178 v[4]->set_list (v[3], v[4], v[5]);
179 v[5]->set_list (v[4], v[5], NULL);
180
181 for (z = 5; z > 0; z--)
182  v[z]->Afisare();
183
184 printf ("Alegeti un numar intre 1..5: ");
185 sfirsit = false;
186 do {
187  tasta = getch ();
188  z = tasta - 0x30;
189  v[1]->move_top (z, v[1]);
190  v[1]->Afisare();
191  if (tasta == '0') sfirsit = true;
192 } while (!sfirsit);
193}
```

FEREAST2.CPP

Here comes the music ...

1. INTRODUCERE

În zilele premergătoare Sfintelor Sărbători de Paşti, m-am plimbat cu fiica mea prin centrul oraşului nostru (Timişoara).

Moştenitoarea mea, cu spiritul ei de observaţie deosebit, (moştenit!), a sesizat în magazine, prezenţa unor instrumente jucărie electronice, gen "miniorgă", impunînd "lepuraşului", drept răsplată pentru notele bune de la şcoală, procurarea acestor jucării... Numai că, "lepuraşul" nostru, lucrează din greu la o instituţie bugetară, la care salariul multiplu "liberalizat", de abia satisface nevoile curente ale familiei.....

Dar problema tratată, trebuia cumva repede rezolvată. Fiind pasionat de calculatoare, mi-am adus aminte că, în "limbajul" GW-BASIC, pe care subsemnatul îl stăpîneşte cel mai bine, există comanda:

SOUND <frecvenţă>,<durată>, care produce sunete de frecvenţa <frecvenţă> (valori între 37-32767 Hz) şi durata <durată> (valori între 0-65535 "tacte" (1/18,2 părţi dintr-o secundă)).

De aici, ideea programului ce urmează....

2. PROGRAMUL REALIZAT

M-am gîndit să transform tastatura PC-ului meu ("rablă", fără HD, cumpărat cu banii de maşină, spre disperarea soţiei! (dar în fond, oricum, în viaţa aceasta eu nu-mi voi putea cumpăra DACIA!)), într-o "miniorgă" electronică, care să satisfacă dorinţele Deliei, (fiica mea!). A rezultat programul MUZICA.BAS, prezentat alături.

3. CONCLUZII.

"lepuraşul", anul acesta, a fost "grozav". Deşi ideea acestui program este extrem de simplă, la îndemîna oricui, economiile realizate de "iepuraş" au fost substanţiale, iar bucuria Deliei uriaşă.

Unde mai punem la socoteală faptul că fiica mea se "joacă" permanent cu calculatorul "rablă" al tăticului....

Atenţie tătici! N-o să ajungeţi prea des să lucraţi la calculatorul Dvs., căci el va fi "acaparat" de copii!.

Programul "merge", evident şi pe TIMS-S, şi pe..., cu modificări minore.

În încheiere, tastaţi "melodia":

QQWQQRE QQRQQTR...

Distracţie plăcută!

Ş.l. ing. Marinceu Dimitrie M.Gh.
U.T. Timişoara, Fac. Mecanica

```
10    REM  MENIU ALEGERE NUMAR OCTAVE
20    CLS : PRINT : PRINT " Program realizat de iepuras pentru Delia"
30    PRINT "*****************************************************"
40    PRINT " MENIU ALEGERE OCTAVE"
50    PRINT "*****************************************************"
60    PRINT "   1-0 OCTAVA"
70    PRINT "   2-DOUA OCTAVE"
80    PRINT "   3-PATRU OCTAVE"
90    PRINT "   4-SFIRSIT"
100   PRINT "*****************************************************"
110   INPUT "ALEGEREA TA:"; N%
120   IF (N% 1 OR N% 4) THEN PRINT "GRESEALA!": GOTO 20
130   ON N% GOTO 140, 500, 800, 1000
140   REM PRIMA "SUBRUTINA" MUZICALA!
150   CLS : PRINT : PRINT " CIND DORESTI SA TE OPRESTI, APASA "; 0; "!"
160   PRINT "  TASTE MUZICALE:q,w,e,r,t,z,u,i,Q,W,E,R,T,Z,U,I!"
170   PRINT "  PENTRU DURATE DE SUNET DUBLE, LITERE MARI!"
180   a$ = INKEY$
190   IF a$ = "q" THEN SOUND 130, 2
200   IF a$ = "w" THEN SOUND 147, 2
210   IF a$ = "e" THEN SOUND 165, 2
220   IF a$ = "r" THEN SOUND 174, 2
230   IF a$ = "t" THEN SOUND 196, 2
240   IF a$ = "z" THEN SOUND 220, 2
250   IF a$ = "u" THEN SOUND 245, 2
260   IF a$ = "i" THEN SOUND 262, 2
270   IF a$ = "Q" THEN SOUND 130, 4
280   IF a$ = "W" THEN SOUND 147, 4
290   IF a$ = "E" THEN SOUND 165, 4
300   IF a$ = "R" THEN SOUND 174, 4
310   IF a$ = "T" THEN SOUND 196, 4
320   IF a$ = "Z" THEN SOUND 220, 4
330   IF a$ = "U" THEN SOUND 245, 4
340   IF a$ = "I" THEN SOUND 262, 4
350   IF a$ = "0" THEN GOTO 20
355   goto 180
500   REM ....SIMILAR....DAR MAI MULTE TASTE ACTIVE!
800   REM ....SIMILAR....DAR SI MAI MULTE TASTE ACTIVE!
1000  END
```

MUZICA.BAS

Reversi - 64:0

Este imposibil să fi lansat Windows fără să fi încercat măcar o partidă de "Reversi". Iar dacă aţi început să pierdeţi mai mult decît să cîştigaţi, de la un anumit nivel în sus, aveţi acum ocazia să vă "răzbunaţi" crunt, administrîndu-i calculatorului o "bătută" cum n-a mai văzut.

Să nu mai pomenim de impresia pe care o lăsaţi asupra prietenilor (care încă nu s-au abonat la "if") dacă reţineţi pe de rost succesiunea de mişcări care vă asigură o victorie de 64:0 (!!).

Nu uitaţi să alegeţi nivelul de dificultate dorit; soluţiile prezentate mai jos presupun că vă imaginaţi tabla de joc cam ca o tablă de şah; colţul din stînga sus este A1, cel din dreapta jos H8. La începutul jocului, cîmpurile D4 şi E5 sînt acoperite de piesele roşii.

Michael Kett/Bernd Nies

```
c5 - c6;  f3 - d2;  b1 - a7;  g5 - pass;
c7 - b5;  h3 - b6;  a8 - b7;  e7 - g7;
a5 - c4;  d1 - c2;  b8 - e6;  h8 - g8;
d3 - b4;  a6 - f5;  e8 - g4;  f8 - pass;
b3 - a4;  c1 - f2;  h4 - h5;  f7 - You
a3 - c8;  e1 - d8;  h6 - h2;  Won by
f4 - c3;  e2 - f1;  h1 - pass;64;
d6 - e3;  g1 - a2;  g6 - f6;
d7 - g3;  a1 - b2;  g2 - h7;
```

Solutia pentru "Novice"

```
c5 - e8;  g8 - f7;  f2 - f1;  b4 - a4;
f5 - g4;  d8 - g6;  g1 - d2;  h8 - a2;
f4 - f3;  h5 - h6;  b1 - c3;  a1 - pass;
h4 - h3;  h7 - g2;  c1 - e1;  b5 - pass;
h2 - c4;  d7 - e3;  d1 - pass;a3 - pass;
g3 - c6;  c7 - c8;  b7 - b6;  a4 - You
g5 - f6;  b8 - pass;a6 - a7;  Won by
d6 - e7;  d3 - c2;  a8 - g7;  64;
e8 - f8;  h1 - e2;  b3 - b2;
```

Solutia pentru "Expert"

În contra viruşilor

Dacă folosiţi în mod curent programul SCAN (McAfee) pentru controlul antivirus (versiunea cea mai nouă: SCAN89) adăugaţi următoarele semnături în SCAN.EXT:

"FA 2E 8C 1E 72 05 BC C8 07"	EDCL
"61 24 FC E6 61 2E C6 06 1F"	Eddie-1800 (Dark Avenger xx?)
"FA 2E 8C 1E 8D 0E BC 7C 15"	Glue (Lipici, Timi$oara, ...)
"50 1E 06 B4 25 B0 01 0E 1F"	Dracula

Iar dacă vă interesează colaborări serioase în ce priveşte organizarea unei "reţele" naţionale pentru protecţia sistemelor şi prevenirea/combaterea efectelor viruşilor, iată o adresă care vă va fi fără îndoială de folos: HotSoft S.R.L., C.P.172, O.P.1, 4300 Tîrgu Mureş.

```
1   {program: doctor.pas
2   autor: Helmut Schuhmacher}
3
4   uses Crt, Dos;
5   type P = array[0..511] of Byte;
6   var Z: array[0..39,1..9,0..1] of ^P;
7     N: array[0..39,1..9,0..1] of Boolean;
8     q,w,u,k,o: Byte;
9     R: Registers; c,d: Char;
10    J: Integer;
11    A: Boolean;
12  function Y: Byte;
13   begin
14    R.ah:=1; Intr(19,R); Y:=R.ah;
15   end;
16  procedure X(q,w,u,k,o: Byte; var B: P);
17   begin
18    R.ah:=q;
19    R.al:=1;
20    R.dl:=w;
21    R.dh:=o;
22    R.ch:=u;
23    R.cl:=k;
24    R.bx:=Ofs(B);
25    R.es:=Seg(B); Intr(19,R)
26   end;
27
28  begin
29  d:='r'; A:=True; ClrScr;
30  Write ('Discdoctor: 720 sect. = 360 ko');
31   GotoXY(1,3);
32   for u:=0 to 39 do
33   for k:=1 to 9 do
34   for o:=0 to 1 do
35   begin
36    N[u,k,o]:= false;
37    Write(Ord(N[u,k,o]))
38   end;
39   repeat
40    begin
41     A:=d='w';
42  A:=NOT A;q:=3-Ord(A);
                          GotoXY(1,15);
43     WriteLn(d,': Drive(0=A, 1=B)? ');
44     ReadLn(c); Val(c,w,J);
45     GotoXY(1,3);
46  for J:=0 to 32767 do Port[$3f2]:=48+w;
47    end;
48    begin
49     for u:=0 to 39 do
50     for k:=1 to 9 do
51     for o:=0 to 1 do
52      begin
53       if N[u,k,o]= NOT A then
54       begin
55        if A then New(Z[u,k,o]);
56        X(q,w,u,k,o,Z[u,k,o]^);
57        if Y=0 then N[u,k,o]:=A
58          else begin
59           Dispose(Z[u,k,o]);
60           X(0,w,u,k,o,Z[u,k,o]^);
61          end;
62       end;
63       Write(Ord(N[u,k,o]));
64      end;
65      WriteLn;
66    end;
67    GotoXY(1,18);
68    WriteLn('DISK: read, write, end');
69    ReadLn(d) until d='e'
70  end.
71
```

DOCTOR.PAS

Print - o comandă ignorată?

Deşi este una din comenzile vechi ale sistemului DOS, surprinzător de puţină lume ştie să se folosească eficient de această comandă. Ea permite listarea în "background" la imprimantă, utilizatorul PC-ului fiind liber să facă altceva în timp ce "curge" listarea. Mai ales la listările care durează mult - vezi orice listare care merge în regim grafic pe o imprimantă matriceală - folosirea acestei posibilităţi se recomandă cu prisosinţă. Chiar dacă acest lucru înseamnă un mic efort suplimentar - cum ar fi de exemplu listarea prealabilă a unui fişier mai mare în mai multe fişiere mici pe disc, pentru a putea relua convenabil listarea în cazul în care se întîmplă ceva cu vreo paginâ.

Un doctor pentru dischete

Programul alăturat serveşte la regenerarea unor dischete greu citibile, care iniţial au fost scrise corect dar apoi au ajuns să provoace deseori erori la citire, fie datorită unor manipulări greşite, fie datorită unor zgîrieturi, cîmpuri magnetice, căldurii, frigului, îndoirii sau pur şi simplu datorită decalibrării unităţii floppy. Deasemenea, programul poate fi util la citirea unor dischete care au fost presate şi care nu pot fi salvate decît după scoaterea lor din învelişul deteriorat şi introducerea lor într-un alt înveliş, în bună stare, care să permită rotirea dischetei. Cu ajutorul acestui program veţi putea, în multe cazuri, să citiţi corect dischete pe care poate că altfel le-aţi abandona. "Doctorul" funcţionează după următoarea metodă:

discheta cu probleme poate fi citită de oricîte ori se doreşte. Sectoarele citite odată corect rămîn în memoria de lucru a calculatorului şi nu trebuie recitite. La repetarea citirii nu vor fi citite decît acele sectoare a căror citire corectă încă nu a reuşit; se repetă acest lucru de cîte ori este nevoie pentru a reuşi citirea dischetei.

Discheta problemă poate fi între timp plimbată şi pe cealaltă unitate (dacă dispuneţi de două unităţi floppy). După fiecare tură, programul arată cîte sectoare au fost citite corect. Dacă toate sectoarele au fost citite corect sau a "n"-a citire nu mai aduce nici un fel de îmbunătăţire, puteţi scrie toate sectoarele citite pe o altă dischetă, după care aveţi o copie bună a dischetei originale. Indicaţii de utilizare: lansaţi "doctorul" apelînd programul. Pe ecran veţi putea urmări 720 de zerouri - cîte unul pentru fiecare sector. Fiecare 0 semnifică starea unui sector de pe dischetă, şi anume faptul că (încă) nu a putut fi citit corect. La citrea corectă a unui sector, zero-ul corespunzător lui se preschimbă-n "1". După prima parcurgere a dischetei, se repetă interogarea asupra următoarei acţiuni: (r- citire, w- scriere, e- ieşire). Pentru scriere se recomandă o dischetă pe care aţi formatat-o în prealabil fără erori. De data aceasta, trecerea cifrelor afişate din 1 în 0 semnifică faptul că sectorul corespunzător a putut fi scris fără erori. Desigur, şi scrierea u puteţi repeta de un număr oarecare de ori, dacă este necesar. Deoarece programul - unul din candidaţii la premiul pentru programele ce ocupă cel mult 1024 ko în sursă, la colegii noştri de la DOS International - a renunţat la orice interogare suplimentară de siguranţă ("Sigur vreţi să scrieţi pe discheta aceasta?"), este recomandabil să vă feriţi de posibilităţi de greşeală, protejînd la scriere discheta pe care doriţi să o refaceţi.

(Helmut Schumacher/I.F.)

Cod	Nume pachet	Nr. dischete	Format	Capacitate	Limba
Sisteme de operare					
29	DR - DOS 6.0	1	3,5"	1,4M	E
Proiectare circuite electronice					
41	Addi-Data: Signal Tools	1	5,25"	1,2M	E
42	MicroSim Corporation: Design Center Demo	1	5,25"	1,2M	E
43	ORCAD	1	5,25"	1,2M	E
44	ORCAD: PC Board Layout Tools	1	5,25"	360K	E
Aplicaţii					
12	CAPP	1	5,25"	1,2M	D
2	Buch,Trio	2	5,25"	360K	D
39	BALPRO - balanţă contabilă	1	5,25"	360K	R
40	UNO - completare formulare	1	5,25"	360	R
45	Hamor Soft: hMISS - secretariat, hCONT - contabilitate, hGEN&hUTIL - dezv. aplicaţii	1	5,25"	360	R
Utilitare, biblioteci					
14	Borland Windows Tools	1	5,25"	360K	D
17	KAO BTX-Dekoder + Demo	1	5,25"	360K	D
36	QuickStep Tools	2	5,25"	1,2M	D
6	Spindrift Libraries Demo	1	5,25"	360K	E
7	BKS Demo	1	5,25"	360K	D
38	List v 3.0	1	5,25"	360K	R
Baze de date					
10	Paradox 3.0	1	5,25"	360K	D
13	Foxpro 1.02	2	5,25"	360K	D
31	Foxpro 2.0	2	3,5"	720K	D
18	SOS - Software Tools Datenbank	1	5,25"	360K	D
24	Advanced Revelation	1	3,5"	720K	E
25	Rbase 3.1	1	3,5"	720K	D
32	WindowBase	1	5,25"	1,2M	D
33	Address One	1	5,25"	360K	D
37	Superbase 4	2	3,5"	720K	D
Grafică, DTP, CAD					
11	Time Works Publisher	2	5,25"	360K	D
15	MEGA CAD	1	5,25"	360K	D
27	Aldus Page Maker 4.0	1	3,5"	720K	E
30	Draw Perfect 1.1 si Office 3.0	1	3,5"	1,4M	D
9	Micrografx	1	5,25"	360K	D
Cursuri					
5	DOS - Curs IBM	1	5,25"	360K	D
Procesoare de texte, OCR					
28	Word Perfect for Windows	1	3,5"	1,4M	D
35	Scout OCR	1	5,25"	360K	D
40	Ekta Editor V1.5	1	5,25"	360K	E
Măsură şi control					
3	Labtech Notebook Demo	1	5,25"	1,2M	E
4	Labtech Control	1	5,25"	1,2M	E
34	Scope	1	5,25"	360K	D
Jocuri, diverse					
8	Pearl Agency	1	5,25"	360K	D

Un nou serviciu pentru cititorii noştri:

SHARE - if

Pentru o mai bună circulaţie a informaţiei între ofertanţi şi solicitanţi de programe (şi nu numai ...), vă oferim posibilitatea să:

- economisiţi timpul necesar introducerii programelor mai lungi, al căror listing a apărut în "**if**".
- preluaţi pe dischetă, pentru eventuale multiplicări, textul articolele din "**if**" pe care aţi vrea să le aveţi disponibile şi în această formă
- preluaţi acele programe şi texte de articole care ne-au fost trimise spre publicare, dar pe care, din variate motive (cel mai adesea, lipsa de spaţiu) - nu ajungem să le publicăm. (Desigur, aceste materiale le difuzăm numai atunci cînd există acordul autorului - încă nu am găsit o variantă de recompensare a autorului pentru această variantă de difuzare) O listă incompletă: criptare fişiere, completare formulare, o parolă solidă, nucleu de multitasking, "catalog" de dischete, etc.
- faceţi cunoştinţă cu acele programe din care ni s-au trimis demo-uri, respectiv **să difuzaţi programele dumneavoastră demonstrative** prin intermedierea noastră. Acest din urmă punct sperăm să intereseze în mod deosebit pe toţi autorii noştri de programe, cărora le este destul de greu să-şi facă publicitatea de care au, evident, nevoie.

Cei dornici să includem demo-urile lor în oferta noastră sînt rugaţi să achite, odată cu expedierea dischetei sursă, suma de 500 lei în contul nostru: Cont nr. 40729-96010402, Banca Română pt. Dezvoltare Mureş.

(Cei care trimit materiale informativ/didactice, şi nu publicitare, nu au de achitat nimic.) Preţul pe care îl practicăm pentru o dischetă este, din păcate, greu de definit în avans, din motive cunoscute de toată lumea. De aceea vă rugăm să vă interesaţi - telefonic sau în scris - în momentul lansării comenzii. În principiu, preţurile sînt identice cu preţurile cu care, la rîndul nostru, am achiziţionat dischetele, la care se adaugă 50 lei pentru munca noastră şi, suplimentar, taxa de expediere. Desigur, cei care ne trimit dischetele lor, au de achitat doar taxa de copiere şi expediere, iar cei care trec sau trimit pe cineva pe la redacţie, vor avea de achitat doar cei 50 lei, taxa de copiere.

Vă rugăm ca pe comanda pe care ne-o trimiteţi să specificaţi pe lîngă numele pachetului şi capacitatea şi formatul dischetelor pe care doriţi să vă furnizăm produsul.

SYSTEM PLUS

este distribuitor autorizat pentru toata gama produselor

DELL COMPUTER CORPORATION

si

HEWLETT PACKARD

NR.1 IN PC-URI DIN SUA SI EUROPA NR. 1 IN FIABILITATE

In acelasi timp va oferim si solutia unui furnizor unic pentru:
RETELE NOVELL si UNIX, SOFTWARE, PERIFERICE, ACCESORII, FAX, UPS
de la cele mai renumite firme americane si japoneze:
EPSON, MICROTEK, SUMMAGRAPHICS, HOUSTON INSTRUMENT, MICROSOFT, BORLAND, CLARIS, ALDUS s.a.
Asistenta tehnica, scolarizarea si sevice-ul in colaborare cu
LAB. DE ELECTRONICA FUNCTIONALA, FACULTATEA DE ELECTRONICA, I.P.B.
tel/fax (90)791391, (90)108342

Talon de comandă-abonament

Subsemnat(a/ul).....
doresc să-mi trimiteţi |_____| exemplare din fiecare număr al revistei **if**,
începînd cu numărul |_____| , pe adresa:
.....
.....
.....
Telefon: acasă..... la servici.....
Pentru aceasta am expediat prin mandat poştal/virament/cec.....lei în contul nr. 40729-96010402 deschis la Banca Română pt. Dezvoltare Mureş. Abonamentul funcţionează pînă la epuizarea sumei; în cazul că banii pe care i-am trimis nu mai ajung pentru un alt număr, voi fi anunţat de acest lucru în timp util, astfel încît nu risc să pierd nici un număr. Dacă mă răzgîndesc şi renunţ la abonament, mi se returnează toată suma rămasă.

if 3 / 92 **Data:** ..... **Semnătura** .....

"if" nr. 4/92 va apare pe piaţă în jurul datei de 1 iulie 1992!

Important !

În curînd va apare cartea

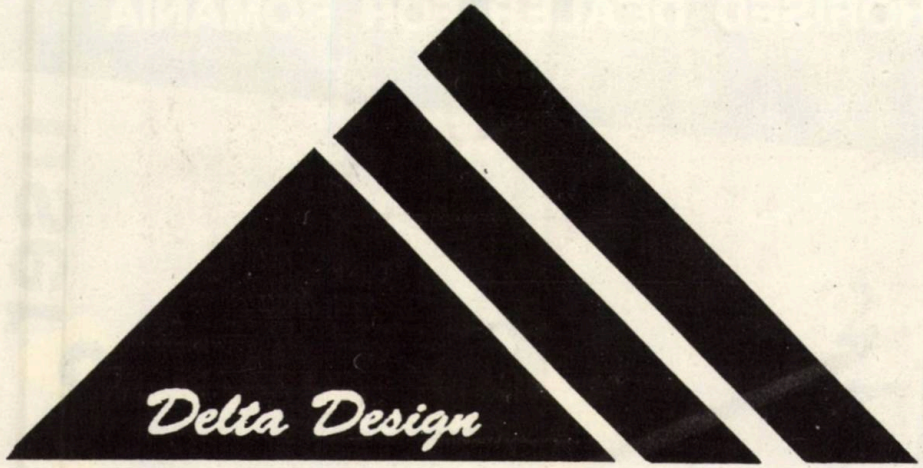
Programarea în limbajul Pascal

editată de Micro ATCI Tîrgu-Mureş.

Manualul este o traducere din limba germană şi este destinat atît celor care vin în contact pentru prima dată cu acest limbaj de programare cît şi celor care doresc să se perfecţioneze învăţînd noi metode şi tehnici de programare. Volumul este bogat în exemple, toate programele prezentate fiind încercate şi testate cu ajutorul compilatorului Turbo Pascal 6.0 al firmei Borland.

Tirajul fiind limitat, asiguraţi-vă că nu veţi pierde lucrarea expediind o comandă fermă pe adresa:

Micro ATCI, C.P. 64, 4300 Tîrgu-Mureş



Delta Design S.A., Hotel Bucureşti, 63-81 Victoria Street,
Wing D - Agencies, Stair E 2, 4th Floor, Sect. 1, Bucharest,
Phones: 15.42.26 / 40 015.45.80 Ext.2404 / Fax: 40 13.60.40.

99.94% of the users fully satisfied of the delivered products

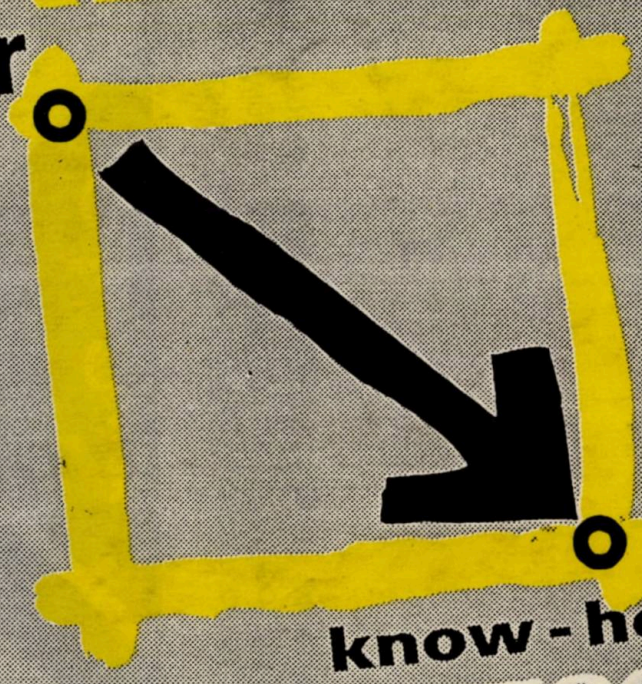
Authorised distributors for the following companies:

Tulip Computers, Apple Computer Inc., Microsoft, Autodesk,
Ashton Tate, Claris, Corel, Delta Design International etc.

AUTODESK AUTHORISED DEALER FOR ROMANIA

The shortest way between

CAD user



the know-how in AUTOCAD

is passing through

雅仕 A&C INTERNATIONAL S.A.

Tel: 53.53.15
Fax: 12.77.74 Telex: 10674 MAXIM R
Technical Support - Service "Hot Line": 12.77.73

if

CALCULATOARE

PERSONALE

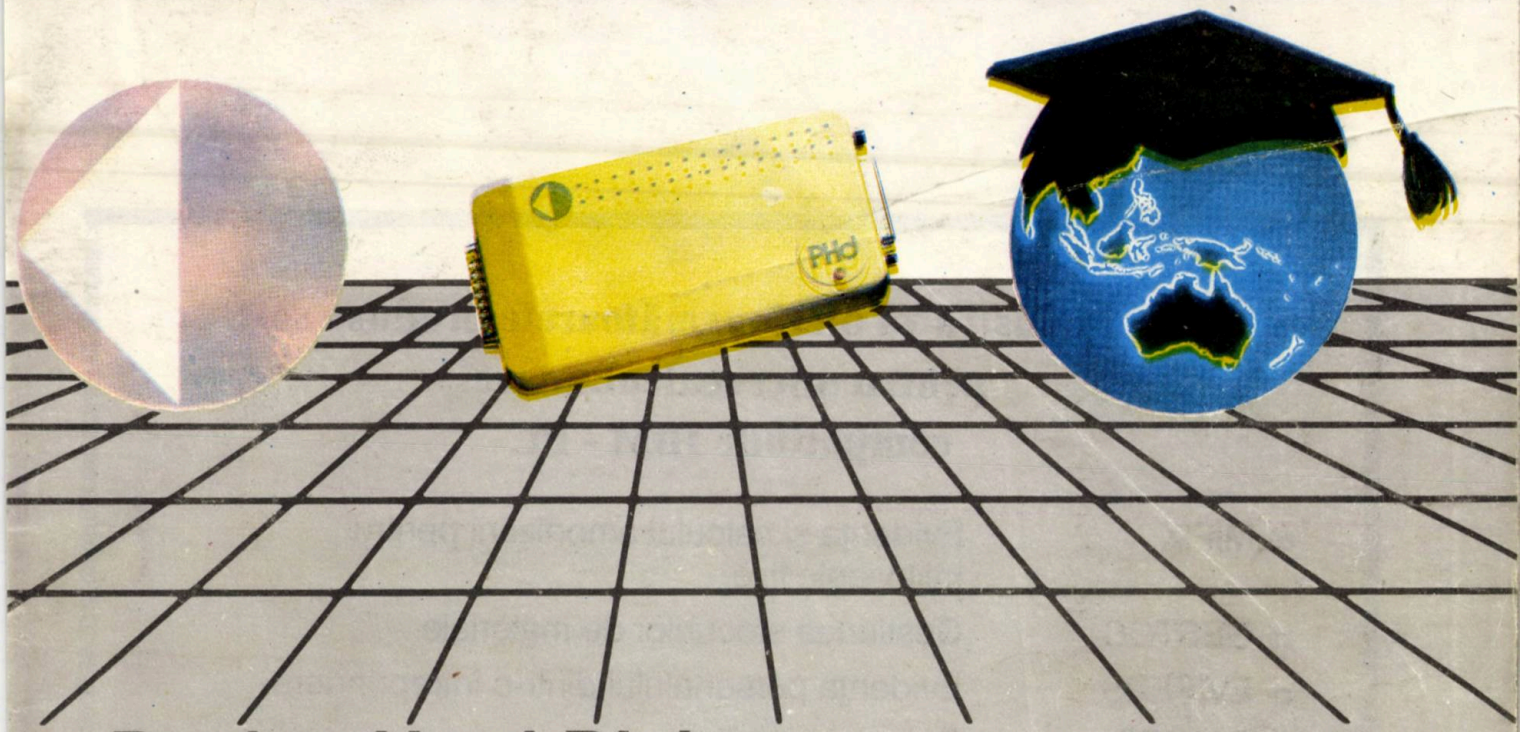
Revistă editată de Micro ATCI S.R.L. Tîrgu-Mureş

5 / 92

(16)

150 lei

ISSN 1220-1529



Pocket Hard Disk

★ Affordability ★ Portability ★ User Friendly ★ Security

★ Simplicity ★ 40 MByte to 120 MByte formatted capacity

...and able to withstand shocks of 150G"

PHd is easily connected to any personal computer parallel port, providing a trouble free, hard disk drive installation.

PHd performs as the next logical drive on the host system, allowing the user to use his own software and data anywhere a computer is available. Once stored, the disk can be removed ensuring absolute data security.

With **PHd**, managers, financiers, engineers and service personnel can carry all their computer files with them.

With **PHd** any computer screen will look like your own.

μmicro ATCI S.R.L.

str. Gheorghe Doja 36
C.P. 64, O.P. 1
4300 Tîrgu-Mureş
Tel./Fax: 954 - 31660

Firma noastră vă oferă următoarele produse soft pentru microcalculatoare compatibile IBM - PC

▫ MIFIX	Evidenţa şi calculul amortizării pentru mijloacele fixe
▫ GESTOC	Gestiunea stocurilor de materiale
▫ EVIPERS	Evidenţa personalului dintr-o întreprindere
▫ SALARII	Calculul salariilor
▫ BALANŢA	Balanţa contabilă şi "Cartea Mare"
▫ CASA	Întocmirea registrului de casă
▫ BANCA	Gestiunea operaţiunilor cu banca
▫ REPARAŢII	Planificarea şi urmărirea reparaţiilor utilajelor dintr-o întreprindere
▫ METR	Evidenţa verificării pentru aparate de măsură şi control
▫ ISCIR	Evidenţa verificării pentru instalaţii sub incidenţa ISCIR

Produsele sînt realizate sub sistemul de gestiune a bazelor de date PARADOX Versiunea 3.5

Editorial 5

Noutăţi 6

OS/2

OS/2 2.0 A meritat aşteptarea 10

De ce OS/2? 10
Ce este nou în OS/2 2.0? 10

OS/2 Versiunea 2.0

Lupta pentru locul la fereastră 14

C.A.C.

Maşina de minciuni PC 16

Interviu

INTERVIU Bill Gates 20

Fundamente

Metoda Backtracking 22

0.Pretext 22
1.Punerea problemei 22
2. Rezolvarea problemei 22
3. Algoritmul Backtracking 22
4. Probleme rezolvabile cu ajutorul metodei Backtracking 22

Reţele Neuronale 26

Ce este analiza senzitivă? 26
Programul exemplu 27
Analiza rezultatelor 28
Pentru dezvoltări ulterioare 29

Reţele

Administrarea reţelelor 37

Crearea de utilizatori şi securitatea 37
Gestionarea aplicaţiilor 37
Salvări 38
Listări 38
NetWare 38
LAN Manager 39
LAN Server 40
Care este cel mai bun? 41

ARTISOFT O revoluţie în conectarea calculatoarelor 42

Baze de date

Un tabel de control pentru succes . . 45

Prima şi a doua generaţie DMS 45
Problemele tipice ale generaţiei a II-a a sistemelor DMS 46
Soluţiile oferite de cea de-a doua generaţie 47
Direcţii de orientare: abordare practică . . . 50
Rezumat 51

Hardware

Flash stîrneşte senzaţie 52

O privire asupra procesării Over Drive 54

Laborator

Utilizarea mouse-ului în modul grafic 320x240x256 56

Tips & Tricks

Copierea rapoartelor în PARADOX . . 59

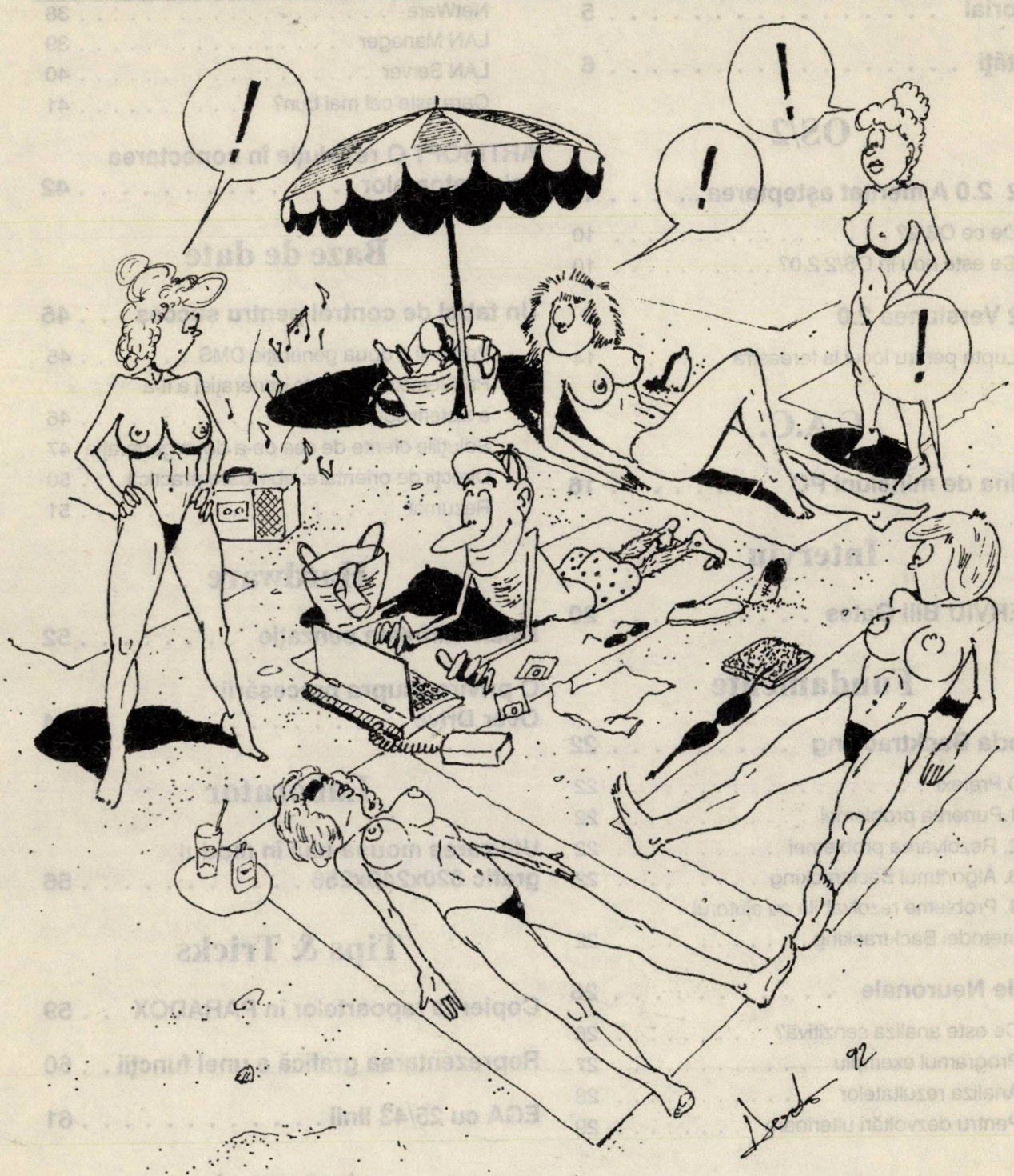
Reprezentarea grafică a unei funcţii . . 60

EGA cu 25/43 linii 61

Anunţuri

SHARE - if 62

PROPUNERE 62


!
!
!
!
92

if

revistă de informatică

editată de firma

Micro ATCI

Director:

ing. Dumitru Dunca

Redacţia:

ing. Cristian Nagy
Doina Ceşa

Colaboratori externi:

stud. Vasile Antonescu
ing. Mihai Beer
ing. Cristian Cosma
mat. Ioan Cozac
ng. Octavian Cudalbu
mat. Eugen Rotaru
stud. Tiberiu Socaciu

Tiparul:

tipografia Tîrgu-Mureş

Tiraj: 4000 ex.
Preţ: 150 lei

Manuscrisele originale sau listingurile de programe sînt primite cu plăcere cu condiţia să nu fi fost publicate în altă parte. Prin expedierea unui manuscris pe adresa redacţiei autorul consimte implicit la publicarea materialului său în revistă. Onorariul se negociază cu redacţia.

Materialele nepublicate nu se restituie.

În sfîrşit, un nou număr din "if"... Sper că, aşa cum sugerează titlul unuia din articolele acestui număr, a "meritat aşteptarea". Care (ea, aşteptarea) a fost - trebuie să recunosc - destul de lungă.

Mulţumesc tuturor celor care nu ne-au uitat. Şi, avînd în vedere că ochii care nu se văd se uită, mă voi strădui ca de acum înainte să ajung la o frecvenţă mai mare a apariţiilor. Eu mi-aş dori să apărem lunar. Vom face totul ...

Numărul de faţă insistă ceva mai mult asupra versiunii 2.0 a celebrului sistem de operare OS/2 elaborat de IBM. Presa spune că, deja, OS/2 nu mai are nevoie de "presă proastă". Cele două articole despre OS/2 intenţionează să constituie numai un fel de prezentare a acestui sistem de operare pentru calculatoare personale. Deşi nu este strict legat de tema OS/2, vă recomand să citiţi incitantul şi incisivul interviu cu William Gates - preşedintele firmei Micrososft. Veţi vedea ce înseamnă "presă prostă" pentru OS/2.

Cele două articole grupate sub genericul "Fundamente" sînt foarte interesante, deşi par a se adresa mai ales teoreticienilor. Totuşi, poate vor interesa şi pe alţii.

Despre reţelele de calculatoare veţi întîlni două articole. Primul face o comparaţie între serviciile oferite administratorului unei reţele (ştiu, ei, administratorii de reţele cum poate fi viaţa uneori ...) de către trei importante sisteme de operare. Cel de-al doilea articol este o prezentare a firmei ARTISOFT şi sistemului de operare pentru reţele locale numit LANtastic. Poate vă va stîrni curiozitatea.

Din "Laborator", colegul nostru Nelu Cozac continuă să vă dea informaţii despre modul X al interfeţei VGA (320 x 240 x 256). Acum a trecut la şoricei.

Ţin să vă atrag atenţia asupra caricaturii de pe pagina 4: este a doua dintr-o serie de desene foarte reuşite care ne-au fost trimise de un viitor matematician: studentul Valentin Antonescu.

Poate că v-ar trezi interesul şi o privire asupra anunţului de pe ultima pagina. Aşteptăm cu nerăbdare reacţiile dumneavoastră - evident, sper că veţi fi entuziasmaţi.

Dar cum orice lucru are un capăt (în afară de cîrnat care are două, după cum spune un proverb arab), închei această introducere cu un îndemn "mobilizator" primit de la unul din cititorii noştri: Go For "if" !!!

Cristian Nagy

Sistemul informaţional al Olimpiadei Barcelona '92 bazat pe OS/2

În cazul unui eveniment ce se desfăşoară la o scară atît de grandioasă, ca Jocurile Olimpice de vară, obţinerea informaţiilor în timp util este un serviciu de maximă importanţă pentru sportivi, crainici, reporteri, spectatori şi membrii de familie. Inima sistemului informaţional olimpic de la Barcelona, 1992, a constituit-o sistemul de operare OS/2 al lui IBM.

Sistemul informaţional pentru rezultate, sistemul pentru comentatori, sistemul informaţional pentru familia olimpică şi sistemul informaţional pentru conducerea operaţiunilor s-au bazat pe două calculatoare gazdă mainframe IBM 9021 şi peste 4400 de staţii de lucru PS/2 lucrînd sub OS/2. De acest sistem informaţional olimpic a depins satisfacerea cerinţelor a peste 15000 de sportivi, 11000 de reprezentanţi de presă, 40000 de membri ai "familiei olimpice" şi 30000 de voluntari.

"Un sistem informaţional de o asemenea mărime ridică probleme deosebite, inclusiv servirea a peste 96000 de utilizatori cu un grad de instruire în computere foarte diferit" spunea Fernand Sarrat, director adjunct cu probleme de dezvoltare a pieţei de sisteme personale de la IBM.

OS/2-ul este proiectat pentru a rezolva aceste probleme şi pentru a gestiona cereri suplimentare ale utilizatorilor, cum ar fi procesare locală, multitasking şi conectivitate impecabilă între cele patru sisteme primare.

OS/2 integrează într-un mediu unic uneltele necesare dezvoltării aplicaţiilor complexe ce compun sistemul informaţional olimpic.

Sistemul informaţional pentru rezultate, cu 400 de calculatoare PS/2 legate în reţea, a fost instalat în fiecare bază sportivă. El permite introducerea manuală sau automată a rezultatelor competiţiilor şi furnizează aceste rezultate în diferite formate, cum ar fi legătura directă cu calculatoarele agenţiilor de presă sau afişări pentru crainicii TV.

Sistemul informaţional şi de comunicaţii olimpic, bazat pe 2000 de PS/2-uri legate în reţea, asigură ziariştilor, arbitrilor, sportivilor şi voluntarilor accesul la informaţii generale cum ar fi rezultatele întrecerilor, programarea concursurilor, situaţia medaliilor, informaţiile pentru presă şi buletinele meteorologice. Acest sistem asigură un serviciu de poştă electronică (E-mail) pentru membrii familiei olimpice şi pentru presă.

Sistemul pentru comentatori, bazat pe 1100 de PS/2-uri legate în reţea, permite comentatorilor de radio şi televiziune să aibă acces la informaţii cu privire la evenimente sau la biografiile sportivilor prin intermediul unui touch-screen color. După îmbunătăţire ca -viteză şi volum- faţă de precedenta Olimpiadă, la care comentatorii aveau acces numai la cîteva canale de comunicaţie simultan.

Sistemul informaţional pentru conducerea operaţiunilor, bazat pe un computer AS/400 de putere medie şi 350 de PS/2-uri legate în reţea, este utilizat de funcţionarii olimpici pentru a gestiona cazarea, rezervarea de bilete, transportul şi serviciile medicale.

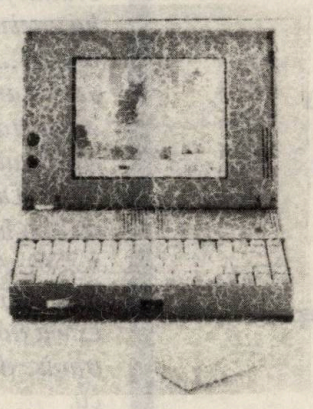
Inovaţiile sistemului informaţional olimpic includ:

- disponibilitatea instantanee a datelor şi rezultatelor de la toate locurile de concurs.
- transmiterea imediată a rezultatelor neoficiale.
- identificarea automată a limbii în care doreşte utilizatorul să primească informaţiile.

IBM Eastern Europe

Acorn pune un RISC într-un notebook

Acorn Computers a anunţat primul său notebook, realizat în jurul procesorului RISC de 32 de biţi ARM care rulează la viteza de 12 mips. Firma că maşina are aceeaşi putere ca şi Power Book 170 al lui Apple, dar la jumătate de preţ.



Modelul A4 de bază, care costa 1399 £ cu 2MbRAM şi disc flexibil de 2Mb şi 1699 £ cu disc hard de 60Mb, se adresează utilizatorilor din învăţămînt şi oamenilor de afaceri, precum şi celor 200000 de utilizatori Acorn.

Notebook-ul utilizează sistemul de operare RISC OS3 al lui Acorn. Directorul de producţie al companiei Henrg Howarth, a confirmat că maşina va rula aplicaţii DOS sub emulare la o viteză care o face echivalentă cu un procesor 286.

"Maşina este capabilă să ruleze şi aplicaţii Windows, dar performanţele ar putea fi mult mai bune", a adăugat el.

(Acorn)

HP pretinde că este pe primul loc

Un studiu - sponsorizat de HP - asupra siguranţei în funcţionare, efectuat în 1991, pretinde că PC-urile HP sînt de peste două ori mai sigure decît PC-urile IBM şi de trei ori mai sigure decît cele Compaq. Studiul, efectuat în SUA de către International Data Corporation, a chestionat 500 de companii despre cinci furnizori de PC-uri, inclusiv AST şi Dell.

Dell, clasat pe locul întîi în ceea ce priveşte satisfacerea clienţilor într-un studiu efectuat de Data Pro/Computer WeeKly în Marea Britanie şi într-un studiu efectuat de JD Power în SUA, a fost sceptic cu privire la rezultate. "Oricît de departe ar merge studiile imparţiale şi prestigioase despre satisfacţia clienţilor, noi sîntem vîrful" spune Maurice Cowley, directorul cu vînzările de la Dell.

Şi IDC s-a interesat de satisfacerea clienţilor, realizînd un clasament cu şapte companii. Pe locul întîi s-a situat HP cu punctajul 6,6 urmat de Compaq cu 6,3. Dell a venit pe ultimul loc cu 6,1. În schimb, Dell a ieşit pe primul loc la repararea PC-urilor cu o medie a ratei de returnare de 2,9 zile faţă de 3,6 zile pentru HP. Principala cauză a defectării a maşinilor o constituie harddiscul, 50% din reparaţiile PC-urilor Compaq au fost generate de probleme cu harddiscul.

CA achiziţionează Nantucket

Casa de soft Computer Associates International a anunţat că a încheiat achiziţionarea lui Nantucket, firma care a elaborat sisteme de dezvoltare a aplicaţiilor compatibile dBase/XBase.

Softul Nantucket va fi vîndut cu etichete CA. Computer Associates a spus că va lua măsuri de asigurare a continuităţii pentru calitatea asistenţei tehnice, productivitatea salariaţilor şi dezvoltarea produselor pe perioada integrării lui Nantucket în companie. Preţul pentru Clipper va rămîne 595 £.

(Computer Associates)

Criză de discuri flexibile

Un val imens de cerere de software a produs o criză de discuri flexibile de 3,5 in, criză ce ar putea să dureze şi o bună parte a anului viitor.

Vina este dată pe lansarea simultană a lui Windows 3.1 şi OS/2 2.0. Windows ocupă şapte dischete de 3,5 in, în timp ce OS/2 utilizează 21. În numai şase săptămîni s-au vîndut circa trei milioane de copii de Windows şi o jumătate de milion de copii de OS/2.

Un purtător de cuvînt al lui Maxell, unul din cei mai mari furnizori de discuri flexibile, a confirmat criza, adăugînd că nu se aşteaptă la o îmbunătăţire a situaţiei generale pînă în anul viitor şi că preţurile dischetelor vor creşte semnificativ.

(PC Direct)

Cyrix lansează un procesor ce va face să scadă preţurile

Se estimează că preţurile PC-urilor se vor prăbuşi odată cu lansarea clasei de procesoare Cyrix de 32 de biţi pentru instrucţiuni 486 care vor constitui primul atac real la adresa dominării pieţei de 486DX de către Intel.

Chip-ul 486DLC se poate instala pe o placă de bază standard pentru 386DX (cu cîteva modificări minore în BIOS şi cîteva modificări hard care să permită 1K de memorie cache pentru procesor). Dacă producătorul doreşte, se poată instala pe placa de bază şi un soclu pentru un circuit de dublare a frecvenţei de ceas. Noul 486DLC are bus de 32 de biţi dar, spre deosebire de 486DX, nu are coprocesor matematic.

Procesorul Cyrix 486DLC de 25MHz şi coprocesorul matematic însoţitor Cx87DLC se vor vinde cu 119$, versiunile de 33MHz cu 155$, iar versiunile 40MHz cu 199$ (circa jumătate din preţul unui 486DX de 33MHz). Noul Cx486DLC lucrează la 3 Volţi şi este în întregime static.

Directorul cu marketingul pentru Europa de la Cyrix, Kelly Van De Hey, afirmă că chip-urile vor determina încheierea monopolului Intel asupra lui 486DX şi-l vor determina pe gigant să-şi reducă preţurile. El spune: "Între preţurile pentru 486SX şi 486DX există o distanţă prin care ar putea trece şi un camion. Intel nu are de ales: fie scade preţurile, fie pierde o parte din piaţă".

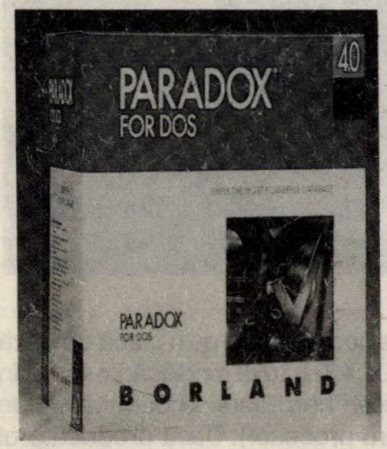
Cyrix afirmă că Tondon, Tandy, Wyse, CompuAdd şi Zeos intenţionează să construiască maşini bazate pe noul Cx486DLC.

Borland aplică noua tehnologie WARP la Paradox 4.0

La conferinţa ţinută în iunie la Marriot Hotel, în căldura toridă din Palm Desert, Philippe Kahn a făcut o demonstraţie cu versiunea 4.0 a cunoscutului Paradox.

Încorporînd ceea ce Borland numeşte noua tehnologie WARP (Wildly Accelerated Relational Performance), Paradox 4.0 este în prezent, produsul cu cea mai mare viteză de pe piaţă. În testele din cadrul demonstraţiei s-a dovedit a fi, în medie, de zece ori mai rapid decît predecesorul său (versiunea 3.5) şi a reuşit să depăşească confortabil performanţele lui FoxPro 2.0, recenta achiziţie a lui Microsoft.

Modificările cosmetice includ o interfaţă cu utilizatorul complet nouă, cu ferestre care pot fi dimensionate şi poziţionate în orice loc pe ecran. "Borland Desktop"-ul permite utilizatorului să vadă mai multe tabele de date simultan şi, pentru prima dată, este incorporat suport complet pentru mouse.



Cu ajutorul mouse-ului utilizatorul poate apela meniuri pentru a vedea şi selecta opţiuni ale programului, sau poate muta coloanele tabelelor.

Îmbunătăţirea manipulării datelor include noile şi mult doritele cîmpuri memo de lungime variabilă care pot conţine pînă la 256Mb de informaţie pentru un cîmp şi maximum 4Gb pentru o tabelă. Cîmpurile memo pot fi create cu uşurinţă cu ajutorul unui editor încorporat, iar interogările şi căutările se pot face pe toate cîmpurile memo.

Cei care folosesc procesoare de texte cum ar fi WordPerfect pentru crearea de documente se vor bucura aflînd că datele memo pot fi asociate unor programe externe. Fişierele create de alte programe pot fi folosite ca date, iar un dublu click pe datele aflate într-o tabelă încarcă şi lansează editorul cu care au fost create.

Produsul poate manipula şi noul tip de date cunoscut sub numele de Borland Blobs (Binary Large Objects) care poate conţine orice tip de date binare cum ar fi imagini statice, animaţie şi fişiere audio. Această caracteristică a fost demonstrată la conferinţă utilizînd imagini şi sunet de la ultimele Jocuri Olimpice, smulgînd vii aplauze din partea publicului. Borland a explicat că acestea au fost posibile datorită noii tehnologii Interbase, o nouă şi puternică metodă de tratare a datelor complexe.

Facilităţile de lucru în reţea şi de acces la date partajate au fost îmbunătăţite cu o gestionare mai bună a încuierii fişierelor. Din nefericire, sistemul nu este compatibil cu versiunea precedentă.

Alte noi caracteristici include extinderi ale PAL-ului care permit utilizatorilor să creeze meniuri, ferestre de dialog, butoane de selecţie şi liste de selecţie. Application Workshop permite crearea interactivă de meniuri şi ecrane cu ajutorul mouse-ului.

Paradox 4.0 va fi disponibil la preţul 499,95#. Un pachet LAN, care adaugă un utilizator suplimentar la reţea va costa 239,95#. Utilizatorii înregistraţi cu versiunea 2.0 sau superioară îl pot obţine contra 129,95# iar cei cu "Borland Passport" cu 79,95#. Versiunea runtime va costa 149,95#.

La conferinţă a fost prezentat şi un cod beta pentru Paradox pentru Windows care va fi disponibil în circa două luni.

(PC World - August 1992)

Dell oferă Novell 3.11

Dell a început să instaleze "din fabrică" Novel Netware 3.11 pe server-ele şi pe staţiile sale de lucru utilizînd o tehnologie automatizată dezvoltată împreună cu gigantul softului de operare pentru reţele.

Capacitatea de integrare va reduce costurile suportate de clienţi pentru integrarea în reţea şi va îmbunătăţi poziţia vînzătorului direct în ceea ce priveşte furnizarea de soluţii pentru file-server - este de părere Jeremy Welch, director de marketing pentru produse la Dell.

Aceasta este prima situaţie cînd NetWare este instalat automat, conform specificaţiilor utilizatorului, înainte de livrare. "Acest fapt oferă o opţiune foarte atrăgătoare pentru utilizatori" a comentat Graeme Alan, director de marketing tehnic la Novell.

(PC Direct)

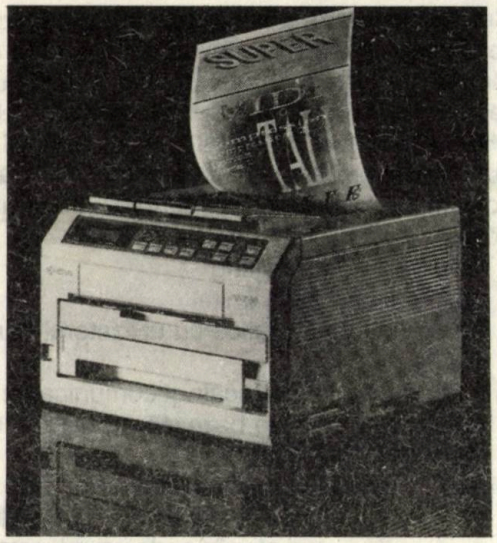
Viteză fără coprocesor

Firma Sigma Design a dezvoltat o nouă interfaţă VGA pentru calculatoare compatibile IBM PC. Noua placă sigma Legend GX este construită cu un nou chip accelerator S3-GUI care aduce pe ecran reprezentările grafice cu o viteză superioară circuitelor vga sau Super VGA. Astfel, noua interfaţă îmbunătăţeşte performanţele aplicaţiilor care utilizează GUI, cum ar fi Windows 3.0.

Interfaţa răspunde şi cerinţelor CAD oferind rezoluţii de 1280x960 şi 1280x1024 puncte în 16 culori simultane.

Imprimantă Kyocera cu cilindru amorf

În sfîrşit ceva nou în domeniul imprimantelor laser! Kyocer a elaborat o nouă metodă de construcţie pentru cilindrul unei imprimante, susţinînd că va reduce considerabil preţul listării unei pagini. Noua tehnologie pentru cilindru este ceramică - siliciu amorf, sau mai exact, a Si. Secretul costului scăzut per pagină constă



în faptul că un cilindru din a Si este extrem de durabil (durata sa de viaţă este aceeaşi cu a imprimantei).

ECOSYSFS-1500 (prima imprimantă care foloseşte a Si) are şi alte avantaje: ea poate utiliza hîrtie reciclată fără deteriorarea cilindrului şi emite cantităţi foarte reduse de ozon.

Din nefericire, Kyocera nu va dezvălui tehnologia altor fabricanţi înainte de a obţine maximum de beneficiu. Ceea ce ar fi în regulă dacă FS-1500 ar fi o imprimantă grozavă; dar nu prea pare să fie.

Mai întîi părţile bune. Poate lista 10 pagini pe minut, ceea ce este rezonabil şi suportă emulare HP LaserJet III, HP-GL2 şi HP7475A. Are doi conectori JEIDA pentru adăugare de plăci cu fonturi. Are un port serial şi un port paralel (care pot recepţiona date simultan) şi poate să aibă opţional-interfaţă AppletalK, Ethernet sau token ring.

Partea proastă este că preţul de 3450$ şi un procesor relativ slab: 68000 la 16 MHz cu maximum 5Mb de RAM. Singura opţiune PostScript este o dezvoltare ROM bazată pe o versiune Kyocera de PostScript Level 1. Opţiunea este relativ ieftină (620$), dar nu include RAM suplimentar pe placă sau procesor suplimentar.

De asemenea, tehnologia unicat pentru cilindru leagă utlizatorul de cartuşele de toner Kyocera.

Ethernet într-un singur chip

AMD a anunţat o implementare Ethernet într-o singură capsulă care poate fi montată direct pe placa de bază a unui PC. El va fi disponibil la un preţ sub 25$ pentru comenzi peste 10000 de bucăţi. Se aşteaptă apariţia maşinilor echipate cu acest chip în trimestrul al treilea al anului.

Circuitul este compatibil cu cunoscutele plăci NE2100 şi NE1500T şi cu softurile de reţea Novell, LAN Manager, Banyan Vines LANtastic.

Conform afirmaţiilor AMD, circuitul are un consum mai mic decît soluţia cu placă de interfaţare şi este cu 20-30% mai rapid.

(AMD)

IBM face o surpriză detractorilor lui OS/2

IBM i-a surprins pe observatorii care i-au considerat nule şansele de a avea succes cu mult întîrziatul sistem de operare OS/2 2.0 anunţînd că se aşteaptă să vîndă între două şi patru milioane de exemplare ale programului pînă la sfîrşitul lui 1992.

După cum susţine IBM, 90% din vînzările deja comandate prin linia directă permanentă de vînzări din SUA au mers la utilizatori de Windows.

(OS/2 User Group)

Unitate de disc flexibil categoria uşoară

Noua unitate de disc flexibil produsă de Anca Associates este mică şi uşoară - ea măsoară numai 50,8x70x12,7 mm şi cîntăreşte numai 75g. Capacitatea de stocare este de 42,6Mb cu un timp mediu de acces la sector de 5,5 ms şi cu un consum de 0,15W în Sleep mode.

Unitatea poate fi instalată într-un notebook, într-un PC uzual, într-o imprimantă, într-un aparat medical, etc. Acest drive Single Disk de 1,8 inch costă circa 400$.

(Gigasys)

OS/2 2.0

A meritat aşteptarea ...

În 1987, IBM anunţa OS/2 ca platformă de elită pentru calculatoarele personale ale sfîrşitului deceniului '80 şi mai departe. Ca parte a arhitecturii aplicaţiilor pentru sisteme, OS/2 a devenit elementul cheie al strategiei software generale a lui IBM.

În 1992, la apariţia lui OS/2 versiunea 2.0, OS/2 rămîne piatra unghiulară a strategiei de software pentru desktop a lui IBM. Şi aceasta deoarece numai OS/2 răspunde celor mai vaste cerinţe ale momentului şi va corespunde noilor cerinţe ce vor surveni către mijlocul deceniului '90 şi ulterior.

Preşedintele IBM-ului, Jack Kuehler, spunea cu ocazia anunţării OS/2-ului:

"Nevoia de OS/2 de piaţă rămîne mai mare ca niciodată. Pur şi simplu, clienţii noştri au nevoie de un sistem de operare care să le permită să investească cu încredere în viitoarele lor aplicaţii. OS/2 continuă să fie strategia pe termen lung a IBM-ului".

De ce OS/2?

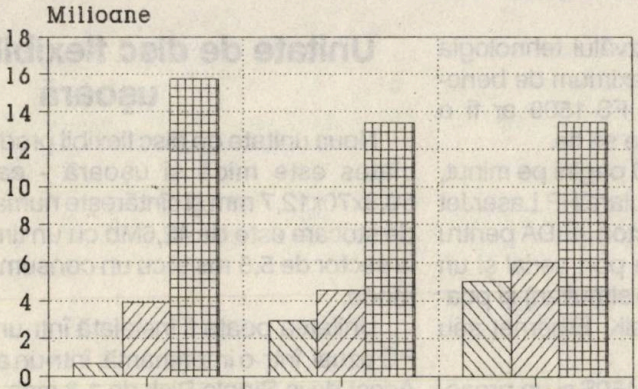
Cel mai popular sistem de operare pentru calculatoare personale din ziua de azi, DOS - care este în continuare un sistem de operare pe 16 biţi, nu a fost proiectat să exploateze clasa de sisteme cu Intel 80386 şi I486. DOS a fost proiectat pentru aplicaţii cu un singur utilizator şi pentru interconectare limitată. Bazele proiectării sale nu mai corespund pentru următoarele motive:

- nu permite multitasking
- spaţiu de memorie limitat
- nu are interfaţă grafică cu utilizatorul (fără Windows)
- integrarea în reţele locale şi comunicarea cu calculatoare gazdă se bazează pe utilizări dincolo de posibilităţile sale.

MS-Windows versiunea 3.0 a extins funcţiile mediului DOS, depăşindu-i unele limitări - mai ales în ceea ce priveşte asigurarea de memorie suplimentară şi multitaskingul limitat pentru aplicaţii DOS. Totuşi, Windows nu este un sistem de operare, ci o extensie construită deasupra lui DOS - deci are nevoie de DOS şi este supus multor limitări ale acestuia. Deşi Windows oferă un anumit grad de multitasking, aplicaţiile DOS sau Windows care nu funcţionează corect pot afecta alte aplicaţii aflate în curs de execuţie sau chiar sistemul de operare. Aceasta poate impune o nouă lansare a sistemului adesea cu pierderea datelor.

Deci nici DOS, nici Windows - în forma lor actuală - nu pot răspunde cerinţelor anilor '90. Este nevoie de proiectare nouă. Un sistem de operare modern are protecţie, multitasking şi interconexiune incluse de la început nu adăugate ulterior.

Vînzările de DOS. OS/2 şi Windows 1991 - 1992

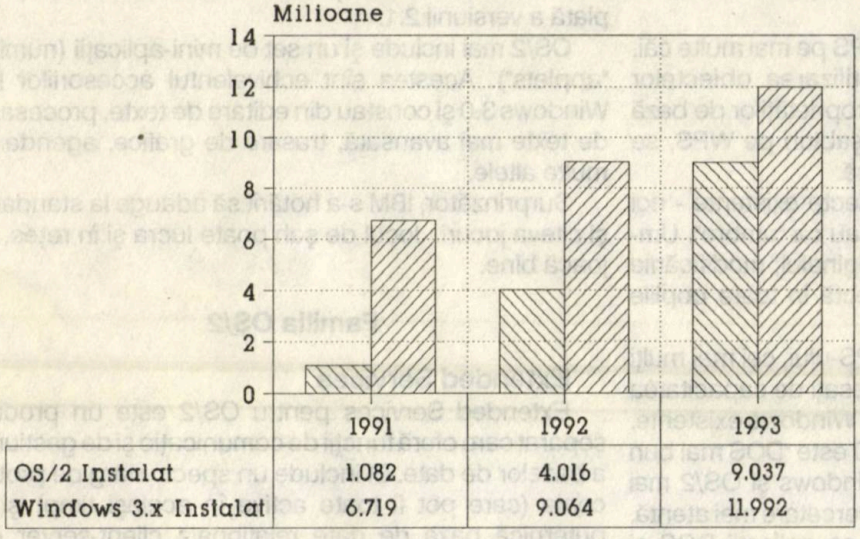


	1991	1992	1993
Total OS/2	0.69	2.934	5.021
Total Windows 3.0	3.919	4.532	5.996
DOS	15.75	13.403	13.423

Ce este nou în OS/2 2.0?

Pentru început, s-ar părea că proiectanţii lui OS/2 2.0 au ţinut cont, cu adevărat, de nevoile bene-

Instalări OS/2 în lume faţă de Windows



	1991	1992	1993
OS/2 Instalat	1.082	4.016	9.037
Windows 3.x Instalat	6.719	9.064	11.992

ficiarilor. IBM a realizat aceasta elaborînd versiuni preliminare ale produsului; acestea au fost disponibile aproape pentru oricine le-a vrut - în total peste 130 000. Nu s-a mai întîmplat pînă acum ca dezvoltarea unui produs să fie obiectul unui asemenea scrutin popular.

Pentru fiecare nouă versiune, IBM a "ascultat" reacţia venită nu numai din sursele oficiale, ci şi din reţele de poştă electronică cum ar fi Compuserve şi Fidonet. În loc să încerce să convingă utilizatorii să încerce un nou sistem de operare cu soft complet nou, IBM a învăţat să adauge OS/2-ului caracteristici care să protejeze investiţiile în software DOS şi Windows. Pentru a uşura opţiunea, unele aplicaţii (nu toate) vor rula mai repede şi mai fiabil decît sub sistemul de operare original.

OS/2 se instalează peste aproape orice versiune de DOS sau OS/2. Pentru codul de bază este necesar un spaţiu de minimum 15Mb pe disc, crescînd la 30Mb dacă se selectează toate opţiunile. Nu este necesară formatarea discului decît dacă doriţi să beneficiaţi de avantajele oferite de High Performance File System (HPFS) sau de Boot Manager.

Boot Manager-ul ocupă o partiţie de 1Mb pe hard disk şi permite utilizatorului să aleagă orice partiţie care poate fi încărcată. Chiar OS/2 însuşi poate fi instalat pe o partiţie secundară, permiţînd utilizarea partiţiei primare pentru DOS, OS/1.x sau chiar Unix.

Dacă 30Mb vi se pare un spaţiu enorm pentru rularea unui sistem de operare, amintiţi-vă că acesta oferă echivalentul pentru DOS, Windows 3.0, OS/2, help foarte complex, un limbaj de programare procedural, 20 de mini-aplicaţii ("applets"), un sistem de gestiune a fişierelor, un procesor de texte plus un program penrtu construit diagrame.

Cu toate acestea, instalarea completă durează circa 30 de minute, timp în care vi se oferă posibilitatea rulării unei introduceri ce acoperă conceptele de bază.

Odată ce aţi instalat complet OS/2 2.0 vi se oferă opţiunea de a migra aplicaţii existente către noul Work Place Shell (WPS) al OS/2-ului. Chiar din faza beta, sînt corect detectate şi oferite spre selectare cele mai multe aplicaţii de succes.

Work Place Shell

Diferenţa evidentă între OS/2 2.0 şi versiunile anterioare este Work Place Shell. Este extrem de important să i se stăpînească conceptele de bază înainte de a începe să se lucreze serios.

Schimbarea constă în trecerea de la un spaţiu de lucru orientat pe aplicaţie (ca vechiul Desktop Manager de sub OS/2) către unul orientat obiect.

WPS este una din primele aplicaţii care aderă la standardul IBM Common User Access, deşi nu este implementat în întregime. Sub WPS, orice ce se găseşte pe ecran este un obiect. Dacă doriţi să faceţi ceva cu un obiect, îl selectaţi, apoi WPS oferă o listă de acţiuni posibile. Fiecare tip de obiect are propriul set de acţiuni asociate.

De exemplu, pentru a edita un fişier sub vechea paradigmă, trebuia să găsiţi aplicaţia de editare, să o lansaţi, să-i cereţi apoi editorului să deschidă fişierul dorit. Sub WPS, trebuie să găsiţi fişierul de editat, să activaţi lista cu acţiunile disponibile şi să selectaţi Edit. Sau, fişierul poate fi "tras" peste un editor dacă se află cumva unul la îndemînă în acel moment.

Chiar şi culorile şi fonturile sînt tratate ca obiecte sub WPS. Dacă doriţi să schimbaţi fontul unei ferestre WPS, se trece iconul obiectului font dorit peste obiectul dorit şi este "scăpat" acolo.

Dacă unele obiecte din WPS nu mai sînt necesare, este uşor să fie trase grupat sau individual peste obiectul tocător. Greşelile pot fi uşor corectate.

O altă diferenţă importantă între WPS şi vechiul DesKtop Manager constă în modul de tratare a obiectelor minimizate. Sub Desktop Manager, atunci cînd se minimizează un obiect, el cade pe "masa de lucru" a lui Presentation Manager sau pe cea a lui Windows şi rămîne acolo pînă cînd este selectat sau închis.

Sub WPS, obiectele pot fi configurate individual pentru o anumită acţiune. Ele pot reveni la pliantul "iniţial"; pot fi plasate în Minimized Windows Viewer (un pliant specializat care conţine numai obiectele minimizate); sau pot fi plasate în partea de jos a "mesei de lucru".

Se pot adăuga noi obiecte la WPS pe mai multe căi. Cea mai uzuală metodă este utilizarea obiectelor şablon. Acestea conţin definiţiile proprietăţilor de bază ale obiectelor. Dacă se trage un şablon pe WPS, se poate crea un obiect din acea clasă.

În alt mod, se pot crea din obiecte existente - noi obiecte ca entităţi independente sau ca "umbre". Umbrele au aceleaşi proprietăţi ca originalul; modificările făcute asupra originalului se reflectă în toate copiile umbră.

În pofida tuturor avantajelor WPS-ului, cei mai mulţi utilizatori vor fi, probabil, mai interesaţi de capacitatea OS/2-ului de a rula aplicaţii DOS şi Windows existente. Afirmaţia făcută de IBM că OS/2 2.0 este "DOS mai bun ca DOS, Windows mai bun ca Windows şi OS/2 mai bun ca OS/2" stă în picioare şi la o cercetare mai atentă.

OS/2 efectuează multitasking ca aplicaţii DOS şi Windows (pentru mod real şi standard) în sesiuni separate şi protejate. Aplicaţiile Windows pentru modul enhanced pot rula dacă sînt cu adevărat DPMI (interfaţă cu modul DOS protejat).

Alte proprietăţi ale modului Windows enhanced (multitasking DOS şi paginarea memoriei) sînt asigurate direct de OS/2. Este asigurat serviciul standard DDE (schimb dinamic de date), deşi numai între aplicaţii Windows.

Spre deosebire de Windows 3.0, chiar după încărcarea driver-elor pentru reţea, se pot pune la dispoziţie 712Kb de memorie pentru fiecare sesiune DOS virtuală. Sesiunile DOS individuale au peste 40 de parametri care pot fi configuraţi de utilizator - acoperind toate aspectele privind memoria, driver-ele de dispozitiv, reîmprospătarea video, etc. De asemenea, fiecărei sesiuni i se oferă pînă la 32Mb EMS, 16Mb XMS şi/sau 512Kb de DPMI. Aplicaţiile care rulează în ferestre DOS pot folosi moduri text sau moduri grafice pînă la rezoluţia driver-ului de dispozitiv video.

Mai mult, în rarele cazuri cînd emularea DOS făcută de OS/2 nu este adecvată, utilizatorilor li se oferă posibilitatea de a crea imagini care să poată fi încărcate pentru orice sistem de operare pentru modul real. Este posibil să se ruleze simultan sesiuni separate cu DR DOS 6.0, CP/M-86, DOS cu Desquiew, DOS 3.3, DOS 4.0 şi DOS 5.0. Se pot rula 240 de sesiuni DOS independente.

Suportul pentru DOS şi Windows este mai mult decît satisfăcător pentru a putea considera OS/2 2.0 o platformă utilizabilă. Singurul aspect îngrijorător îl constituie viteza, mai ales cînd se rulează aplicaţii Windows.

Ceea ce este, poate, mai îngrijorător este faptul că mult-aşteptatul instrument grafic pe 32 de biţi pentru OS/2 2.0 nu a apărut la această versiune. De această secţiune a OS/2-ului s-a ocupat partenerul strategic al lui IBM, Micrografx, care a avut întîrzieri serioase. Avînd în vedere că conducerea IBM nu va mai accepta amînări, ne putem aştepta la o lansare destul de apropiată a versiunii 2.1.

OS/2 mai include şi un set de mini-aplicaţii (numite "applets"). Acestea sînt echivalentul accesoriilor lui Windows 3.0 şi constau din editare de texte, procesare de texte mai avansată, trasare de grafice, agende şi multe altele.

Surprinzător, IBM s-a hotărît să adauge la standard şi cîteva jocuri. Jocul de şah poate lucra şi în reţea, şi joacă bine.

Familia OS/2

Extended Services

Extended Services pentru OS/2 este un produs separat care oferă funcţii de comunicaţie şi de gestiune a bazelor de date. El include un spectru larg de protocoale (care pot fi toate active în acelaşi timp) şi o puternică bază de date relaţională client-server cu SQL. Extended Services poate lucra atît cu OS/2 Versiunea 1.3 (16 biţi) cît şi cu OS/2 versiunea 2.0 (32 biţi).

În plus, Extended Services va rula pe calculatoare compatibile IBM PC, nu numai pe PS/2.

OS/2 LAN Server

OS/2 LAN Server este o platformă puternică ce asigură servicii LAN utilizatorilor de DOS, Windows şi OS/2.

Ca şi Extended Services, OS/2 LAN Server poate rula pe calculatoare compatibile IBM PC, ca şi pe PS/2.

DDCS/2

Distributed Database Connection Services/2 (DDCS/2) este un produs complementar lui Extended Services. El oferă unui client OS/2 - conectarea Windows şi OS/2 să aibă acces, lucrînd cu Database Manager, la baze de date de pe calculatoare gazdă în conformitate cu arhitectura bazelor de date distribuite (DRDA), care include DB2, SQL/DS şi OS/400.

Provocarea OS/2

Pentru a face OS/2 să trăiască la adevăratele sale posiblităţi, IBM va trebui să umbrească nu numai strălucirea continuă a lui Microsoft Windows, ci şi a altor rivali. La sfîrşitul lui ianuarie 1992, NeXT Computer anunţa NeXTStep pentru calculatoare Intel 80486, SunSoft a anunţat Solaris iar membrii ACE Initiative se grupează în jurul lui Windows-NT. Constituie toate aceştea o ameninţare pentru OS/2? Examinînd alternativele anunţate, răspunsul este "NU" răsunător. Sistemul de operare NeXT va rula toate aplicaţiile scrise pentru sistemele NeXT - NeXT Cube şi NeXT Stations. Totuşi, există foarte puţine aplicaţii pentru NeXT care

să intereseze pe utilizatorii ne-tehnici. A doua mare problemă este că sistemul de operare NeXT costă de 8 ori mai mult ca OS/2. Sistemul de operare NeXT nu va fi disponibil înainte de trimestrul al 4-lea al lui 1992.

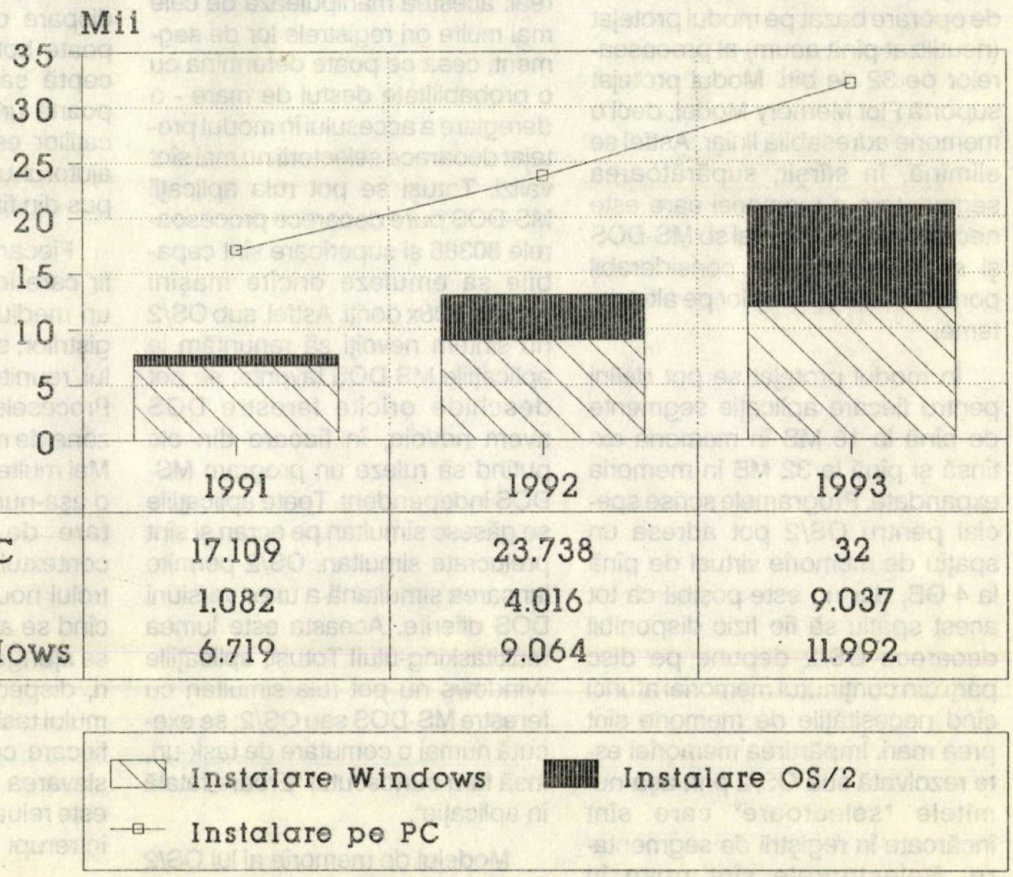
Sun Soft a intrat în luptă cu Solaris, un sistem de operare ce va rula atît pe Sun SPARC cît şi pe procesoarele Intel de 32 de biţi. Din nefericire, Sun a întîrziat cu lansarea. În plus, Solaris este destinat vînzătorilor independenţi de soft care caută un sistem de operare care să le permită să dezvolte atît pentru platformele RISC SPARC, cît şi pentru cele CISC Intel de 32 de biţi.

Cea de-a treia alternativă, iniţiativa ACE cu Windows-NT şi SCO, ar fi putut fi o ameninţare formidabilă pentru OS/2, dar Digital Equipment a blocat progresul acestui grup. Cînd Digital a anunţat că arhitectura lor Alpha/RISC va înlocui sistemele VAX, majoritatea celor două mii de firme independente care dezvoltă soft pentru Digital şi-au mutat interesul de la Windows-NT către OSF/1. Se pare că sistemul de operare strategic pentru Digital şi clienţii săi nu este ACE şi Windows-NT, ci arhitectura Alpha/RISC cu OSF/1.

Cu toată compatibilitatea aplicaţiilor existente cu mediul OS/2 IBM trebuie să-i convingă pe vînzătorii independenţi de soft să scrie aplicaţii pentru OS/2. Este interesant că unii din importanţii vînzători de soft au promis suport. Printre aceştia se numără firme cunoscute ca Lotus, Micrografx, Borland, DeScribe. Pentru a urgenta aceste eforturi, IBM oferă "centre de portare" unde producătorii de soft intră cu aplicaţii DOS şi/sau Windows şi ies cu aplicaţii OS/2 2.0. Timpul mediu necesar pentru portare din DOS sau Windows a fost de trei săptămîni.

Cristian Nagy

Penetraţia Windows şi OS/2 în calculatoarele personale Intel pe 32 de biţi 1991 - 1993



	1991	1992	1993
Instalare pe PC	17.109	23.738	32
Instalare OS/2	1.082	4.016	9.037
Instalare Windows	6.719	9.064	11.992

OS/2 Versiunea 2.0

Lupta pentru locul la fereastră

Versiunea 2.0 a sistemului de operare OS/2 produs de IBM a apărut în aprilie 1992. Această versiune pe 32 de biţi oferă facilităţi de multitasking nemaiîntîlnite în lumea PC-urilor. Lupta dintre IBM şi Microsoft pentru sistemul de operare al viitorului continuă cu şi mai multă violenţă.

OS/2 a fost dezvoltat de IBM pentru a satisface cerinţele în continuă creştere ale hardware-ului. MS-DOS suferă de prea multe limitări - nu permite multitasking, nu suportă memorie peste limita de 1 MByte, etc. Pentru a le elimina, specialiştii de la IBM au creat un sistem de operare bazat pe modul protejat (neutilizat pînă acum) al procesoarelor pe 32 de biţi. Modul protejat suportă Flot Memory Model, deci o memorie adresabilă liniar. Astfel se elimină, în sfîrşit, supărătoarea segmentare a memoriei care este necesară în modul real sb MS-DOS şi se îmbunătăţeşte considerabil portabilitatea aplicaţiilor pe alte sisteme.

În modul protejat se pot defini pentru fiecare aplicaţie segmente de pînă la 16 MB în memoria extinsă şi pînă la 32 MB în memoria expandată. Programele scrise special pentru OS/2 pot adresa un spaţiu de memorie virtual de pînă la 4 GB, dar nu este posibil ca tot acest spaţiu să fie fizic disponibil deoarece OS/2 depune pe disc părţi din conţinutul memoriei atunci cînd necesităţile de memorie sînt prea mari. Împărţirea memoriei este rezolvată sub OS/2 prin aşa-numitele "selectoare" care sînt încărcate în regiştrii de segmentare. Selectoarele sînt puse în legătură cu o structură de date - numită descriptor de segment - care conţin informaţii despre segmente (adresa fizică, mărimea, scopul utilizării, nivelele de prioritate, ş.a.) Programele "văd" numai selectorul, deci nu ştiu, de exemplu, unde anume se află amplasate în memorie la un anumit moment. Descriptorii de segment dînt grupaţi în tabele de descriptori: cea globală, cea locală şi cea de întreruperi. Acest procedeu permite sistemului de operare să mute, la nevoie, un program într-o altă zonă de memorie.

Aici apare o deosebire esenţială faţă de aplicaţiile MS-DOS în mod real: acestea manipulează de cele mai multe ori registrele lor de segment, ceea ce poate determina cu o probabilitate destul de mare - o dereglare a accesului în modul protejat deoarece selectorii nu mai sînt valizi. Totuşi se pot rula aplicaţii MS-DOS pure deoarece procesoarele 80386 şi superioare sînt capabile să emuleze oricîte maşini virtuale 808x doriţi. Astfel, sub OS/2 nu sîntem nevoiţi să renunţăm la aplicaţiile MS-DOS favorite; se pot deschide oricîte ferestre DOS avem nevoie, în fiecare din ele putînd să ruleze un program MS-DOS independent. Toate aplicaţiile se găsesc simultan pe ecran şi sînt prelucrate simultan. OS/2 permite lansarea simultană a unor versiuni DOS diferite. Aceasta este lumea multitasking-ului! Totuşi, aplicaţiile Windows nu pot rula simultan cu ferestre MS-DOS sau OS/2; se execută numai o comutare de task-uri, însă fără cunoscutul "Eroare fatală în aplicaţie".

Modelul de memorie al lui OS/2 pune ordine şi în dereglările de acces care apar prin concurenţa diverselor aplicaţii (cazul programelor TSR (Terminate and Stay Resident) de sub MS-DOS): permite fiecărei aplicaţii să aibă acces numai în zona de memorie care i-a fost alocată. Pentru platforme multitasking serioase acest lucru este indispensabil. Să presupunem că, în timp ce tipărim un document de sub un procesor de texte, un program TSR transmite - aceleiaşi imprimante o altă sarcină de tipărire. Cum poate stabili imprimanta care date au fost trimise de o anumită tipărire. Cum poate stabili imprimanta care date au fost trmise de o anumită aplicaţie? Acest lucru dispare deoarece, sub OS/2, se poate hotărî chiar şi dacă se acceptă sau nu aplicaţii care "se poartă urît". Acest Babilon al aplicaţiilor este rezolvat de OS/2 cu ajutorul unui Tasking Model compus din fire, procese şi sesiuni.

Fiecare task deţine cel puţin un fir care înzestrează programul cu un mediu format din valori ale regiştrilor, stivei şi modul procesorului, reunite sub numele de context. Procesele sînt constituite din fire, zone de memorie, fişiere şi drivere. Mai multe procese sînt reunite într-o aşa-numită sesiune. La o comutare de task, OS/2 salvează contextul programului, predă controlul noului task şi îl rulează pînă cînd se atinge limita de timp. Cînd se ajunge la sfîrşitul listei de taskuri, dispecerul predă controlul primului task dintr-o listă. Deoarece la fiecare comutare de task se face slavarea contextului, fiecare task este reluat din punctul în care a fost întrerupt anterior.

Sincronizarea resurselor este realizată de OS/2 prin intermediul unui dispecer de eşantioane de timp bazat pe o listă de priorităţi, care atribuie tuturor firelor din sistem un eşantion de timp prestabilit; astfel este exclusă posibilitatea unui blocaj al sistemului (dead lock). Dacă - de exemplu - un fir actualizează, la un anumit moment, o structură de date, el nu poate fi întrerupt în acel moment.

În acest scop, OS/2 atribuie o structură specială de date semafor - unui singur fir; semaforul este cerut de fir atunci cînd el face un acces la memorie. OS/2 permite accesul la memorie unui alt fir numai în momentul în care este eliberat semaforul.

Comunicarea între procese este asigurată prin fanioane (flags), cozi de aşteptare (queues) şi canale de comunicaţie (pipes). Cu ajutorul unui fanion un proces semnalizează producerea unui anumit eveniment. Alte procese pot să citească acest fanion cu ajutorul unei funcţii OS/2. Procesele pot transmite date prin intermediul canalelor de comunicaţie (scriere/citire secvenţială în/dintr-un alt proces) sau prin intermediul cozilor de aşteptare (datele pot fi citite în ordinea dorită).

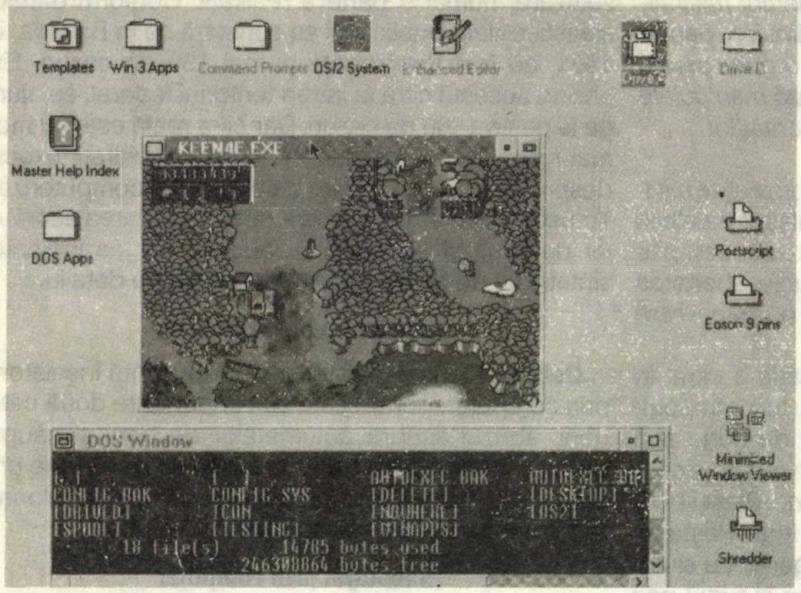
Ca orice sistem de operare, OS/2 trebuie, mai întîi, să citească programele de pe disc. Această funcţie este realizată de o componentă a nucleului: încărcătorul. Atunci cînd încărcătorul primeşte un apel de funcţie _EXEC, el generează un proces şi construieşte structurile de date pentru gestiunea memoriei. Spre deosebire de legătura statică, în care fişierul .EXE trebuie să conţină valori pentru toţi regiştrii segment, OS/2 permite referinţe între segmente care sînt rezolvate numai în momentul execuţiei. Programul executabil conţine numai o referinţă ;a modul care se găseşte în Dynamic Link Library (DLL). Încărcătorul testează dacă această DLL se află deja în memorie şi, în caz contrar, o încarcă.

Fiind un sistem de operare multitasking, OS/2 trebuie să împiedice "jenarea" reciprocă a taskurilor. Această cerinţă este rezolvată utilizînd proprietăţile de protecţie şi izolare a taskurilor în modul protejat al procesoarelor 80X86. Date şi coduri se află la baza unui model de priorităţi cu patru nivele. Cel mai înalt nivel de prioritate (nivelul 0) îl primesc, de obicei, numai funcţiile de nucleu. Astfel, ele au acces la toate datele din sistem. Pe nivelul 1 rulează componentele care asigură servicii de sistem, cum ar fi driver-e de dispozitiv, sistemul de gestiune a fişierelor, etc. Ele au acces la zonele utilizator, dar nu şi la nucleu. Subsisteme cum ar fi tastatura, mouse-ul sau subsistemul video rulează pe nivelul 2. În sfîrşit, nivelul 3 rulează programele utilizator.

Din punct de vedere al aspectului, OS/2 oferă o suprafaţă utilizator grafică asemănătoare cu Presentation Manager, completată în versiunea 2.0 - cu o suprafaţă utilizator orientată obiect numită "Workspace Shell". Aceasta permite tipărirea unui document prin plasarea cursorului mouse-ului, cursorul este înlocuit cu simbolul ales. Ţinînd tasta apăsată, se "trage" simbolul documentului peste simbolul imprimantei şi se eliberează tasta mouse-ului. Acum va începe tipărirea documentului. Acest procedeu este denumit "Drag and drop". La fel se poate proceda şi pentru copierea sau ştergerea fişierelor.

(Traducerea şi adaptarea:

ing. Mihai BEER)



Fereastră DOS în care rulează un joc pe VGA

Este evident că, în viaţa noastră de zi cu zi, calculatoarele ocupă un loc din ce în ce mai important. În unele domenii activitatea nu mai poate fi concepută fără ele. Tot mai multe meserii şi ramuri de activitate devin mai mult sau mai puţin "computer aided..." Şi astfel am ajuns să avem CAD, CAM, CAE, CASE, etc.

De data aceasta vă propunem un domeniu ceva mai nou şi care, mai ales în ultima vreme, începe să se bucure de atenţie deosebită şi din partea farurilor legislative. Este vorba de Computer Aided Crime (CAC) pentru care nu avem - deocamdată! - un echivalent în limba română.Ce aţi spune de ceva în genul "delicvenţa asistată de calculator" ?

În legătură cu acest subiect vă prezentăm un articol publicat în numărul 8/1992 al revistei germane DOS International.

Maşina de minciuni PC

PC-uri pentru spargerea automatelor de jocuri, pentru falsificarea de dovezi şi cifrarea spargerilor mafiote, pentru teletransmisie de date pentru tăinuire, pentru pornografie infantilă, pentru comanda de organe pentru transplant, computere din sisteme de pază manipulate - toate acestea cer atenţie din partea autorităţilor.

Datele săpate în plăci de piatră au supravieţuit mileniilor. Prelucrarea automată a datelor "mişcă" datele în jurul pământului. Oare manipulările şi abuzurile săvîrşite în acest domeniu pot să ameninţe bunurile, libertatea individuală şi statul de drept? Cum pot fi prinşi cei care comit CAC?

Prin noua piesă a codului penal, justiţia a intrat în posesia unui instrument deosebit pentru lupta împotriva abuzurilor prin computer. Dar, oricît de bune ar fi legile, ele ajută prea puţin dacă organele economice şi cele de cercetare penală nu se pot impune. Piaţa liberă nu se prea supune prevederilor acestor legi. "Poşta de spărgători" le ia chiar în derîdere: "spargerea cu ajutorul computerului este acum interzisă, ca şi furtul prin copiere. Dar asta nu deranjează pe nimeni. Numai că pericolul nu constă în faptul că acum fiecare mic spărgător are în spate întreaga putere a statului concentrată ca într-un cartuş, ci în imposibilitatea aplicării legilor de către poliţie care este slab pregătită şi puţin solicitată..."

CAC într-un sens mai restrîns

În legile împotriva infracţiunilor săvîrşite cu ajutorul computerului este vorba, în esenţă, de prevederi privind pătrunderea în baze de date, înşelătoria şi sabotajul, precum şi pirateria software care este reglementată de legea copy-right-ului. Dar aceste delicte, care caracterizează criminalitatea prin computer în sens restrîns, constituie partea cea mai redusă. Ca piraterie software se consideră folosirea nepermisă şi utilizarea fără drept de autor a soft-ului. Cine ascultă plîngerile despre pagubele enorme provocate prin copiere ilegală ar putea crede că cifrele vor exploda. Dar statistica criminalităţii redă, şi în domeniul pirateriei software, numai o parte a realităţii. Conform datelor statistice date publicităţii se constată că în Bavaria, în 1991, au fost înregistrate numai 130 de cazuri. Extinzînd această cifră la întreg teritoriul federal, se ajunge la peste 1500 de cazuri. Dar cifra reală este cu mult mai mare. De asemenea, nu există declaraţii exacte despre aria de cuprindere a spionajului computerizat. Totuşi statistica conţine cifre despre trădarea secretelor de firmă şi ale celor de afaceri, dar ca indicator sintetic ce nu poate permite exprimări de detaliu.

Înşelătoria prin computer

Delictul esenţial din statistici îl reprezintă înşelătoria prin computer, pentru care sînt importante două categorii: abuzul asupra automatelor de bani şi asupra celor de jocuri. Toate celelalte forme de înşelătorie prin computer, despre care circulă zvonuri spectaculoase, sînt conform cifrelor neimportante.

Sabotajul prin computer

Şi statisticile despre sabotajele prin computer arată cît de mare este cifra "la negru". Dacă vom compara numai numărul mare de viruşi răspîndiţi cu delictele de sabotaj prin computer, vom avea oglinda parţială a realităţii. Cauzele se regăsesc, desigur, şi în lipsa de entuziasm a firmelor păgubite de a face publice informaţiile.

În afara statisticilor oficiale asupra criminalităţii, cele mai multe şi mai importante din aceste delicte apar şi într-o statistică internă a "serviciului de informaţii asupra criminalităţii prin computer" al poliţiei.

CAC în sens mai larg

Începînd cu 1985, definiţia CAC sună astfel: "Prin criminalitate prin computer se înţeleg toate faptele în care prelucrarea automată a datelor este un mijloc de acţiune şi care motivează bănuirea unei fapte penale". Această definiţie include atît delictele criminalităţii prin calculator în sens restrîns cît şi toate delictele în care prelucrarea automată a datelor serveşte drept instru-

met - ca de exemplu tăinuirea prin Mailbox-uri. Adversarii acestei definiţii obiectează cum că, odată cu răspîndirea crescîndă a prelucrării automate a datelor, delivenţa prin calculator ar putea deveni - conform acestei definiţii - delictul cel mai frecvent întîlnit în statisticile privind criminalitatea.

Dar exact această sentinţă va trebui să se impună! Aceasta cere funcţionari criminalişti care să fie mai bine pregătiţi în domeniul informaticii, precum şi competenţe poliţieneşti în acest domeniu. Deci informatica ar trebui să devină materie de examen la facultăţile care pergătesc cadre pentru poliţie.

Profesor Klaus Brunnstein, Univesitatea din Hamburg specialist în informatică:

1. Societatea, economia şi statul se bazează din ce în ce mai mult pe integrarea tehnicii de calcul în aplicaţii importante, dar fără a analiza riscurile. Între timp, prosperitatea sau ruina firmelor de stat sau particulare au devenit dependente de tehnica de calcul.

2. Sistemele de calcul actuale, în special PC-urile şi sistemele UNIX, nu au fost proiectate avînd în vedere securitatea şi controlul. Tineri de 16 pînă la 18 ani arată cum se poate abuza, în special prin PC-uri, sau cum pot fi induşi în eroare utilizatorii prin viruşi sau alte softuri. Răspîndirea sistemelor Mac-Intosh şi Unix se confruntă, inevitabil, cu probleme asemănătoare.

3. În ciuda acestor pericole de "spargere" - chiar a reţelelor şi a calculatoarelor mari - noţiunea de criminalitate asistată de calculator este nepotrivită, deoarece posibilităţile de abuz se găsesc, cel mult, în lipsa de securitate dată de chiar proiectarea tehnicii de calcul.

4. Cel puţin tinerii spărgători şi autori de viruşi nu au motive criminale: cum economia apreciază ca pozitivă construcţia, programarea şi introducerea PC-urilor, nu ne putem aştepta ca oameni tineri, fără etică sau cunoştinţe de legislaţie, să-şi considere preocupările "criminale".

5. În plus, sistemele de calcul răspîndite astăzi (PC-uri, reţele) nu conţin componente pentru păstrarea urmelor; sau, în cazul mainframe, ele nu sînt cuplate.

6. În sfîrşit, organele de cercetare (poliţie, procuratură), ca şi experţii în drept (judecători, avocaţi) înţeleg foarte rar elementele criminale ale utilizării sistemelor de calcul. De exemplu, în procesul KGB-ului spărgătorii au fost nevoiţi să explice judecătorului ce înseamnă "Email"; nu este de mirare că profesorul Stoll, un expert în domeniu, nu a putut să convingă completul de judecată cu expertiza sa.

Ţinînd cont de nesocotinţa cu care statul, economia şi societatea s-au făcut dependente de tehnică de calcul nesigură, abuzurile comise - în special de tineri - pot fi cu greu considerate drept criminale şi, în plus, nu pot fi urmărite.

Cauzele diferenţelor între cele două statistici se regăsesc în volumul mic de informaţii de care dispune poliţia şi în faptul că o mare parte a delictelor comise cu calculatorul nu sînt descoperite de poliţie. Cel care analizează cifrele relativ scăzute din statisticile infracţiunilor asistate de calculator poate trage concluzia falsă că pericolul este supraapreciat. Aceasta deoarece cea mai mare şi mai periculoasă parte a acestor delicte, aşa-numita criminalitate prin computer în sens larg, nu apare în statistici. Poliţia - ca şi publicul - nu sînt încă nici pe departe conştienţi de pericol. Este vorba aici de traficul de stupefiante, de comerţul ilegal cu arme, de pornografia infantilă, de diverse forme de delicte economice şi chiar de infracţiuni privind protecţia mediului înconjurător. Din cauza acestui spectru foarte larg, termenul de "Computer Aided Crime" (CAC) a înlocuit - începînd cu sfîrşitul anului 1986 - denumirea de "criminalitate prin computer în sens redus şi lărgit" pentru toate delictele care folosesc prelucrarea automată a datelor ca instrument de acţiune. Abia după clarificarea noţiunilor se înlătură confuzia produsă de cele două statistici paralele ale poliţiei. Şi aceasta deoarece pentru criminalişti şi informaticieni CAC nu este altceva decît un domeniu particular de utilizare a calculatoarelor. În spiritul aceleiaşi viziuni par să acţioneze şi infractorii care înţeleg criminalitatea prin computer drept o "alternativă", cel mult ilegală, de folosire a prelucrării automate a datelor. Exemplele care urmează sînt atît de cunoscute încît direcţionarea subiectului către aspectul ilegal al faptei scapă. Orice cunoscător a diversităţii lumii informaticii şi, în plus, pe cea a criminalităţii şi care mai adaugă şi puţină fantezie îşi poate închipui diversitatea enormă a abuzurilor prin prelucrarea automată a datelor. Chiar "maniacii" calculatoarelor descoperă permanent ceva nou. Cei mai mulţi dintre ei, chiar dacă nu iau în întregime în serios chestiunea furtului prin copiere a softului, sînt totuşi cetăţeni conştienţi ai statului de drept şi pregătiţi pentru a sprijini poliţia în lupta ei împotriva delicvenţei chiar dacă este vorba de trafic de stupefiante, infracţiuni economice grave sau împotriva mediului înconjurător. Cea mai cunoscută aplicare a CAC este comerţul cu programe furate prin Mailbox. Numai în primii ani aceste programe au fost comercializate, în mare măsură, prin anunţuri în ziare. Cînd acest procedeu a devenit prea periculos, făptaşii au început să folosească formulare pentru post-restant. În zilele noastre piraţii deplasează softul prin Mailbox. Noua metodă se poate aplica şi pentru bunuri furate, casete video cu pornografie infantilă sau casete video care glorifică violenţa. Transimisia de date la distanţă prin Mailbox sau Btx reprezintă nu numai calea de comercializare a mărfurilor furate, ci şi cele de comunicare între infractori pe care poliţia nu o mai poate controla. "Peştii mai mici împart viruşi prin Mailbox, unde cel care are box-ul poartă răspundrea. Cît de devreme au abuzat infractorii de mediul Mailbox se poate vedea din faptul că, imediat după instalarea primelor automate de bani, prima indicaţie privind falsificarea Eurocheque-urilor s-a răspîndit printr-un Mailbox spune profesorul Brunnstein care susţine că toate acestea ar putea duce - în viitor - la nenorociri neaşteptate.

Oricum, viruşii computerelor nu sabotează numai operarea calculatoarelor. Printr-o proiectare cores-

punzătoare a părţii distructive pot fi realizate şi delicte de spionaj sau fapte ilegale de şantaj şi constrîngere.

Oricît s-ar proteja prelucrarea automată a datelor prin instrumente de securitate, publicul trebuie să se gîndească la faptul că, pe de altă parte, criminalii pot anula accesul poliţiei la datele care dovedesc efracţia lor. Chiar criminalii dovedesc un interes mult mai mare pentru securitatea datelor decît firmele legale. Acest lucru este lesne de înţeles deoarece dacă un responsabil cu prelucrarea automată a datelor neglijează securitatea datelor riscă, în cel mai rău caz, o sancţiune profesională. Dimpotrivă, un infractor care nu-şi protejează dovezile de ochii poliţiei socoteşte că va fi condamnat.

Credinţa în păstrarea adevărului prin documente scrise - "căci ceea ce deţii scris negru pe alb poţi să duci cu încredere acasă" - a fost zguduită de tehnica tipăririi. De cînd există PC-uri puternice, echipate cu scannere, imprimante laser şi color şi cu software grafic avansat, computerele pot produce "minciuni patentate". Aceasta deoarece documentelor falsificate - indiferent dacă sînt alb-negru sau color - le urmează fotografiile false. Pentru a obţine astfel de succese, astăzi nu mai este necesar - ca înainte - un atelier de falsificat plus o pregătire specială. Prelucrarea numerică micşorează valoarea de dovadă a pozelor sau a negativelor. Digitizarea fotografiilor a două persoane cu ajutorul scannerului, schimbarea figurilor cu un editor grafic, depunerea rezultatului pe hîrtie sau pe dischetă şi transmiterea rezultatului unui atelier pentru executarea unui diapozitiv sau a unui negativ color nu este o îndrăzneală chiar atît de mare. Un nou progres îl aduc aparatele foto cu schiţare digitală.

Nu putem să închidem ochii în faţa dezvoltărilor imaginabile. Computerul este un instrument de falsificare evident. După cum ne învaţă experienţa, infractorii se inspiră fără reţineri din orice nouă dezvoltare hardware. Răufăcătorul care lucra, înainte, la un automat de jocuri cu sîrme, magneţi sau burghie, îşi cumpără astăzi un laptop şi, pentru sume între 5000 şi 10000DM, programe de manipulare.

Jocul criminal de golire a automatelor arată foarte clar cum, prin computerizarea unei infracţiuni care înainte se făcea mecanic, se nasc noi pericole şi cum PC-ul uşurează manipularea criminală. Deşi pînă acum au fost prinşi sute de făptaşi care golesc automatele, nu a fost încă prins nici un autor al softurilor de manipulare.

Exemplele de criminaliate împotriva mediului înconjurător arată pregnant multiplele faţete ale prelucrării automate a datelor. Pe de o parte, numai calculatoarele mai pot garanta controlul prescris al reglementărilor legale de supraveghere a aerului şi apelor, în timp ce, pe de altă parte, deschid înşelăciunii toate căile. Este de la sine înţeles că aici lucrează interdisciplinar biologi, fizicieni, medici şi informaticieni. Dar la funcţionarii criminalişti care asigură probe efectuînd cercetări sau la specialişti informaticieni care să le pregătească nu se gîndeşte aproape nimeni. Rămîne de netăgăduit faptul că nu există securitate sută la sută a prelucrării automate a datelor.

Gunter von Gravenreuth, avocat şi inginer, specialist în procese de piraterie soft:

Chiar dacă pentru poliţie s-au dus vremurile cînd dischetele erau găurite şi apoi fixate ca să nu se piardă ca dovezi, totuşi există şi astăzi mari diferenţe regionale în ceea ce priveşte clarificarea informatică a organelor de cercetare. Aici nu este vorba, precum s-ar crede, de diferenţa dintre sat şi oraş. În timp ce procuratura din Munster, considerat a fin un oraş de provincie, obţine rezultate de vîrf în ceea ce priveşte cercetările, procuratura din Frankfurt vorbea - chiar în octombrie 1990 - despre "un obiect plat de formă pătrată, încă nedescris exact, din material plastic tare", referindu-se la dischetele de 3,5 inch. CAC cuprinde astăzi toate domeniile vieţii comerciale. Ea nu se amnifestă numai în domeniul pirateriei soft. Astfel au fost, de exemplu, oferite mărfuri spre tăinuire prin Mailbox. Accesibilitatea informaticii este folosită pentru şantaj. Înşelătoriile prin computer sînt voalate; se dezvoltă un virus agresiv care devine activ numai pentru produsele concurenţei.

Siguranţa datelor este întotdeauna o problemă de compromis. Prin exagerarea măsurilor de siguranţă cresc costurile care anulează avantajul introducerii tehnicii de calcul. În sens invers, trebuie să se ţină cont cel puţin de condiţiile de bază ale siguranţei prelucrării automate a datelor. Din păcate, în ceea ce priveşte condiţiile de bază, există goluri foarte mari. Nu-ţi poţi mustra secretara că îşi lipeşte parola sub tastatură dacă un mare comerciant de hardware din München livrează reţele la care parola supervisorului este tipărită pe etichete şi beneficiarul nu este atenţionat să o modifice.

Avea dreptate profesorul Nagel de la IBM care, acum 20 de ani, susţinea că noţiunea de criminalitate prin computer este greşită pentru că discreditează informatica. Nici măcar inteligenţa artificială nu produce "computere criminale". Criminal poate fi numai omul, căci numai el converteşte cele mai bune lucruri în arme.

Şi totuşi, nici un calculator nu garantează crima perfectă, dar unele ilegalităţi devin mai eficiente. În plus, computerul permite cifrarea informaţiilor împiedicînd accesul organelor de cercetare la ele, posibilitate intens exploatată de infractori.

Cei care, luînd în considerare statisticile, consideră infracţiunea asistată de calculator ca fiind inofensivă ar trebui să ţină seama de faptul că, în continuare, computerul va preface toate formele de delicvenţă. Această evoluţie nu afectează numai structura organelor de cercetare penală, ci atinge întreaga economie şi chiar statul. Organizaţiile criminale complexe nu pot fi imaginate fără suportul informaticii. Poliţia trebuie să ţină cont de acest fapt în cercetările sale.

(Traducerea şi adaptarea: ing. Mihai Beer)

Trei nume cu rezonanta pe piata internationala, reunite de un singur distribuitor:

POLICOM S.A.

Produse oferite din stoc:

TOSHIBA

- copiatoare
- telefax-uri
- laptop-uri
- imprimante laser
- consumabile

star

- imprimante matriciale (A4 si A3)
- imprimante laser
- consumabile

Polaroid

- dischete 3,5" si 5,25"
- filtre pentru monitoare
- filme foto

Se asigura service in garantie si postgarantie.

Relatii suplimentare la magazinul din CLUJ
B-dul 22 decembrie nr. 135
tel. 95-156350 fax 95-152006 telex 31348

Tot aici mai puteti gasi o gama completa de calculatoare compatibile IBM-PC.

INTERVIU

Bill Gates

Vă prezentăm cîteva fragmente dintr-un interviu cu Bill Gates, preşedintele lui Microsoft Corporation. Interviul, publicat în numărul din aprilie 1992 al revistei "Upside", se referă la noul Windows NT, şi conţine interesante şi incisive referiri la OS/2.

Rep.: Cum va fi posibil ca NT să nu "măcelărească" Windows 3.0 sau 4.1 sau orice altă versiune existentă la momentul apariţiei sale?

Gates: NT nu cere hardware suplimentar. Vom livra un NT care are nevoie de 8 megabytes. De fapt, ar putea rula cu 4, dar vrem să fim siguri că este loc suficient pentru ca cele mai puternice aplicaţii să poată rula. În plus, noi nu credem că NT va ajunge - la început - la un procent mare de utilizatori. Pentru cei mai mulţi este suficient Windows peste DOS.

R: Cine vor fi primii utilizatori de NT?

G: În mod sigur, cei care lucrează pe server-e şi maşini RISC. Cei care au cu adevărat nevoie de securitate. Maşinile multiprocesor se vor baza pe NT.

R: Ce scopuri urmăriţi în legătură cu penetrarea lui NT pe piaţă?

G: Pentru primul an, sau pentru primii doi, 15 sau 20 la sută din piaţă. În timp - daţi-ne 4 ani - totul va trece către NT.

R: Şi apoi?

G: Apoi vom aplica strategia familiei din nou. NT va fi produsul "de jos" şi vom avea lucruri uimitoare pe care le vom aşeza "deasupra".

R: Microsoft ar putea dispune de o familie de sisteme de operare instalabile fără compromisuri pe o clasă în continuă diversificare de calculatoare, de la cele ce pot fi

William Gates III

ţinute într-o mînă la super-calculatoarele multiprocesor?

G: Ne-am hotărît să avem două implementări în ideea de a avea această elasticitate. Dacă vom avea nevoie de mai multe implementări, o vom putea face.

R: O formă a lui Windows pe orice... De fapt mizaţi pe faptul că beneficiarii vor fi tentaţi mai mult de protejarea investiţiilor de soft decît de obţinerea de performanţe într-adevăr optimizate?

G: Nu, nu, nu! Scopul nostru este să nu existe nici un fel de renunţări. Atunci cînd introducem stil în Windows, cred că o facem mai bine decît Go. Băieţii noştri sînt mai inventivi. Noi am făcut o treabă mai bună. În orice caz, noi beneficiem de influenţa lui "Hei, Windows este acest produs implementat masiv". Cînd un programator scrie o aplicaţie, are el vreun cîştig? Eu cred că da. Trebuie să nu uităm că toate acestea sînt mijlocite de existenţa unor aplicaţii extraordinare şi de economia softului.

R: Care ar fi aceasta?

G: Costuri fixe ridicate - 50 milioane $ - pentru scrierea oricărei aplicaţii "mari". Apoi, mult mai mulţi bani pentru creerea cererii şi distribuţiei astfel încît să poată fi vîndută masiv.

R: Hai să vorbim despre sisteme de operare în general. Într-un interviu, Jim Manzi spunea că anii '90 vor fi cuceriţi cu aplicaţii, nu cu sisteme de operare. Are dreptate?

G: Nu ştiu exact ce a vrut să spună cu asta. Sistemele de operare joacă un rol mai important ca nicioadată. De exemplu, facilităţile de transmitere a mesajelor vor fi incluse în sistemul de operare. Ideea de a coordona informaţii partajate prin aplicaţii diferite nu se

Maşinile multiprocesor se vor baza pe NT

poate regăsi decît în sistemul de operare.

R: Bine, dar Notes al lui Lotus face o mulţime din toate acestea, dar este o aplicaţie şi nu un sistem de operare.

G: Aşa este. Dar credeţi că fiecare aplicaţie trebuie să inventeze tehnologii de securitate? Credeţi că trebuie să existe seturi diferite de tehnologii de securitate administrate diferit pentru fiecare aplicaţie de pe sistem? Vreau să spun că, pentru o bună parte din ceea ce a făcut Notes, a trebuit să fie pionier într-o categorie de probleme pe care sistemul de operare nu le oferea. O mare parte din Notes - securitate, directoare şi toate acestea vor fi incluse în sistemele noastre de operare şi în cele ale lui Novell. Aceasta este schimbarea care vine. Pentru realizarea acestora, sistemul de operare este piesa centrală. Nu se poate ajunge la o acoperire mare a pieţei pînă cînd nu vor fi incluse aceste facilităţi. Asemănător, uitaţivă la celelalte noutăţi - suport audio, creioane, ecrane cu atingere, comunicaţie fără fir - toate sînt produse soft de sistem.

R: Ar putea deveni sistemele de operare cerinţele anilor '90?

G: Dintotdeauna au fost numeroase competiţii. Ceea ce se întîmplă acum nu este o noutate. Oamenii tind să uite că, atunci cînd a fost introdus sistemul de operare, erau trei sisteme de operare. Şi IBM era în spatele celor trei. Există o dinamică a unui sistem care tinde să o ia înainte. Este clar că aceştia sîntem noi. În mod sigur avem competitori care doresc să piardă mulţi bani. IBM a aruncat o grămadă de bani pe sisteme de operare.

R: Cam două miliarde pînă acum, nu?

G: IBM se situează într-o clasă aparte în ceea ce priveşte cheltuirea banilor în moduri ineficiente.

R: Revistele comerciale sînt pline de poveşti despre planul IBM-ului de a distribui OS/2 utilizatorilor de Windows şi DOS. Crezi că IBM va încerca cu adevărat să o facă?

G: Deoarece ar trebui să-mi plătească un drept de autor, ar fi o manevră foarte scumpă - destul de stupid pentru ei şi un chilipir pentru noi. Drepturile de autor ar trebui să fie semnificative. Dar chiar şi aşa, OS/2 este mare şi lent şi nu prea există aplicaţii pentru Presentation Manager.

R: Dar IBM trebuie să vadă o ocazie care merită să fie exploatată aproape la orice preţ. Şi anume ca OS/2 va ieşi în martie 1992, iar NT numai către sfîrşitul acestui an.

G: Întotdeauna au avut o ocazie. Aceasta nu este prima versiune a lui OS/2; OS/2 a apărut cu mult timp în urmă. Toată lumea ştie ce este OS/2. Credeţi-mă nu este foarte deosebit. Cea mai mare parte a codului este încă pe 16 biţi. Nu este un sistem robust. Nu diferă chiar atît de mult de prima sa versiune. Dar au avut un avans ucigaş. OS/2 a fost anunţat în aprilie 1987. În curînd îşi va sărbători a cincea aniversare.

R: Există vreun scenariu IBM care să te îngrijoreze? Poate IBM să producă o erodare a pieţei pentru voi?

G: Ceea ce utilizează oamenii astăzi este Windows. Aplicaţiile se scriu pentru Windows. Şi mai avem şi alte aplicaţii, ceea ce constituie motivul pentru care se cumpără sisteme de operare. Cred că ar fi destul de bizar dacă OS/2 va dobîndi popularitate. Totuşi, IBM îl poate băga pe gît unor indivizi.

R: Poate IBM să facă vreun fel de software?

G: Viitorul lor în soft este judecat cel mai bine de trecutul lor în soft. Trecutul lor în soft constă în TopView, OfficeVision, versiunea DOS 4.0 plină de erori şi OS/2. Nu văd, în felul în care fac ei soft, nici o schimbare care să facă viitorul diferit de trecut.

Cred că ar fi destul de bizar dacă OS/2 va dobîndi popularitate. Totuşi, IBM îl poate băga pe gît unor indivizi.

Metoda Backtracking

0.Pretext

Deoarece multe probleme se pot rezolva elegant folosind metode şi tehnici de programare adecvată, iar acestea nu sînt cunoscute de cei care se iniţiază în programarea calculatoarelor prin prezentul articol doresc să pun la dispoziţia cititorilor un material referitor la algoritmul Backtracking. De remarcat că mulţi algoritmi recursivi pot fi transformaţi în algoritmi nerecursivi (uneori mai eficienţi) rescriindu-i în spiritul acestei metode.

1.Punerea problemei

Fie n mulţimi finite oarecare $M_1,...,M_n$ avînd $p_1,...,p_n$ elemente:

(1) $|M_k| = p_k, k = 1,n.$

Fără a restrînge generalitatea, putem considera că:

(2) $M_k = \{1,...,p_k\}, k = 1,n$

deoarece putem eventual considera bijecţiile:

(3) $\Psi_k : \{1,...,p_k\} \rightarrow M_k, k = 1,n.$

Vom nota cu:

(4) $M = M1X ... XMn$

şi în mod evident avem:

(5) $|M| = p_1 ... p_n.$

Vom numi soluţie a problemei, un element al mulţimii

(6) $S = \{x \in M : \varphi(x) = 1\},$

unde φ este un predicat n - ar:

(7) $\varphi : M \rightarrow \{0,1\}.$

Se pune problema determinării mulţimii S. O metodă simplă de determinare a mulţimii S este generarea tuturor elementelor mulţimii M (eventual cu ajutorul "principiului contorului") şi verificarea pentru fiecare element a condiţiei $\varphi(x)$.

Această metodă însă nu convine, timpul de execuţie fiind exponenţial (de exemplu dacă $p_1 = ... = p_n = 2$ algoritmul are timpul de lucru O (2^n)).

2. Rezolvarea problemei

Metoda Backtracking urmăreşte să evite neajunsurile expuse anterior. Pentru aceasta vom introduce pe lîngă condiţiile interne legate prin predicatul φ şi nişte condiţii de continuare φ_k astfel încît:

(8) $\varphi_k : M1X ... XMk \rightarrow \{0,1\}$

şi:

(9) $\varphi_k(x_1,...,x_k) = 0$ şi $k < n \Rightarrow$
$\varphi_{k+1}(x_1,...,x_k,x) = 0$
pentru $\forall\ x \in M_{k+1}$

adică φ_k este un predicat de aritate k, care are următoarele proprietăţi: dacă k este fals pe un k - tuplu, atunci va fi fals pe orice k+1-tuplu care are ca prime k componente pe componentele k - tuplului iniţial.

În mod necesar avem următoarele condiţii:

(10) $\varphi_n = \varphi$

şi:

(11) $\varphi_1 = 0$

Să observăm că dacă

(12) $\varphi_k = 1, k = 1, n\text{-}1$

vom genera mulţimea S prin parcurgerea întregii mulţimi M. De asemenea să observăm că dacă

(13) $\varphi = 1 \Leftrightarrow \varphi_k = 1\ \forall\ k = 1, n$

atunci mulţimea soluţiilor coincide cu mulţimea M, adică:

(14) $S = M \Leftrightarrow \varphi = 1$

3. Algoritmul Backtracking

Predicatele φ_k introduse anterior ne sugerează construcţia unei soluţii

(15) $x \in S\ [x = (x_1, ..., x_n), x_k \in M_k, k = 1, n]$

prin determinarea succesivă a lor xk. Dacă $\varphi_k(x_1, ..., x_k) = 0$, atunci scădem pe k, altfel luăm la rînd acei $x_{k+1} \in M_{k+1}$ neverificaţi încă.

Dau în continuare sursa unui program Pascal şi a unuia C care calculează numărul de permutări şi care determină permutările mulţimii $\{1,...,n\}$, unde $n \in N^*$. De asemenea pentru amatorii de recursivitate ofer un al treilea listing care transformă acest algoritm într-un algoritm recursiv.

4. Probleme rezolvabile cu ajutorul metodei Backtracking

a. Generarea permutărilor mulţimii {1, ..., n}.

Numim permutare a unei mulţimi

(16) $A_n = \{1,...,n\}$

o aplicaţie bijectivă

(17) $\sigma : A_n \rightarrow A_n.$

Teorema 1: f : A → A, A finită, atunci:

(18) f injectivă ⇔ f surjectivă ⇔ f bijectivă.

Demonstraţia: se găseşte in [1].

Pe baza teoremei putem construi predicatele φ_k:

(19) $\varphi_k(x_1, ..., x_k) =$

$$= \bigwedge_{1 \le i < j \le k} (x_i <> x_j),$$

adică am pus condiţia ca σ să fie injectivă. Se poate observa că:

$$(20)\quad \varphi_k(x_1,...,x_k) = \varphi_{k-1}(x_1,...,x_{k-1}) \bigwedge_{i=1}^{k-1} (x_i <> x_k)$$

Dar conform (9) putem simplifica predicatul φ_k:

$$(21)\quad \varphi_k(x_1,...,x_k) = \bigwedge_{i=1}^{k-1} (x_i <> x_k)$$

Putem scrie în Pascal o funcţie de evaluare a predicatului:

```
function fi( k : integer; x:vector ) : boolean;
  var
    i:integer;
    b:boolean;
  begin
    b:=true;
    for i:=1 to k-1 do
      b:=b and (x[k]<>x[i]);
    fi:=b
  end;
```

b. Problema celor n dame:

Să se aşeze pe o tablă de şah cu latura de n pătrate, n dame astfel încît ele să nu se ameninţe una pe alta.

Teorema 2: Numărul maxim de dame care se pot aşeza este n.

Demonstraţie: Presupunem că se pot aseza n+1 dame; tabla de şah avînd n linii, prin urmare există în mod evident o linie pe care se află cel puţin două dame, acestea amenințîndu-se reciproc. De fapt teorema este adevarata pentru n = 4.

Notînd cu x_k linia in care se afla dama (e numai una!) de pe coloana k putem scrie predicatul jk:

$$(22)\varphi_k(x_1,...,x_k)=\bigwedge_{i=1}^{k-1}[(x_i <> x_k)\wedge(k-i <> |x_k-x_i|)]$$

Dar conform (9) putem rescrie predicatul φk:

$$(23)\quad \varphi_k(x_1,...,x_k) = \bigwedge_{i=1}^{k-1}[(x_i <> x_k)\wedge(k-i <> |x_k-x_i|)]$$

Am putea defini o funcţie Pascal:

```
function fi (k:integer;x:vector):boolean;
  var
    i:integer;
    b:boolean;
  begin
    b:=true;
    for i:=1 to k-1 do
      b:=b and (x[k]<>x[i]) and
          (k-1<>abs(x[k]-x[i]));
    fi:=b
  end;
```

c. Colorarea grafurilor (vezi [2] cap. III, 2.3)

Numim graf neorientat

$$(24)\quad G = (X,U),$$

unde X se numeşte mulţimea vîrfurilor (nodurilor), iar

$$(25)\quad U \subseteq X^2$$

se numeşte mulţimea muchiilor. Pentru simplitate, dar fără a restrînge generalitatea putem considera:

$$(26)\quad X = \{1,...,n\},$$

deci:

$$(27)\quad |X| = n.$$

Se pune problema definirii unei funcţii

$$(28)\quad f : X \to C$$

care ataşează fiecărui nod $x \in X$ o culoare $c \in C$. Din nou fără a restrînge generalitatea putem considera:

$$(29)\quad C = \{1,..,m\}$$

si deci:

$$(30)\quad |C| = m.$$

Funcţia f definită în (28) trebuie să satisfacă următoarea condiţie:

$$(31)\quad (x,y) \in U \Rightarrow f(x) <> f(y)$$

adică două noduri legate printr-o muchie sînt colorate cu culori diferite. Numărul minim de culori cu care poate fi colorat un graf se numeşte număr cromatic şi se noteaza cu g(G), adică:

$$(32)\quad \gamma(G) = \inf\{m \in N^* / \exists f: X \to \{1,...,m\} \text{ si } (x,y) \in U \Rightarrow f(x) <> f(y)\}$$

În legătura cu colorarea grafurilor se cunoaşte (vezi [3]) următoarea

Teorema 3: Pentru graf neorientat G care corespunde unei hărţi avem

$$(33)\quad \gamma(G) <= 4.$$

Demonstraţia acestei teoreme este grea; de remarcat că pentru demonstraţia completă s-a folosit şi calculatorul.

Vom codifica o funcţie-soluţie în următorul mod:

$$(34)\quad f(i) = x_i,$$

deci o soluţie se va prezenta sub forma unui n - tuplu $(x_1,...,x_n) \in C^n$.

Vom putea defini predicatul φ_k astfel:

$$(35)\quad \varphi_k(x_1,...,x_k) = \bigwedge_{\substack{1 \le i < j \le k \\ (x,y) \in U}} (x_i <> x_1)$$

sau ţinînd cont de (9):

$$(36)\quad \varphi_k(x_1,...,x_k) = \bigwedge_{i=1}^{k-1} (x_i <> x_k)$$
$$(x,y) \in U$$

Pentru relatia (36) putem scrie o funcţie Pascal:

```
function fi(k:integer;x:vector):Boolean;
  var
    i:integer;
    b:boolean;
  begin
    b:=true;
    for i:=1 to k-1 do
      if u(i,k) then
        b:=b and (x[k]<>x[i]);
    fi:=b
end;
```

Lăsăm pe seama cititorului găsirea unor predicate φ_k pentru următoarele probleme:

a. Realizarea tuturor circuitelor hamiltoniene cu calul pe tabla de şah.

b. Să se construiască toate turnurile formate din h cuburi (discuri) de latură (razã) li şi de culoare ci astfel încît să nu stea un cub (disc) cu latura (raza) mai mare pe un cub (disc) cu latura (raza) mai mică şi să nu stea alăturate două cuburi (discuri) cu o aceeaşi culoare.

c. Să se determine toate submulţimile M ale unei mulţimi finite a de numere reale strict pozitive ale elemente au o sumă dată S:

$$(37)\quad S = \sum_{x \in M} x$$

d. Să se genereze toate combinările şi toate aranjamentele unei mulţimi finite {1,...,n}

5.Bibliografie:

- [1] Ministerul Educaţiei şi Învăţământului - "Algebra. Manual pentru clasa a IX-a"
- [2] Leon Livovschi & Horia Georgescu - "Sinteza şi analiza algoritmilor"
- [3] Ioan Tomescu - "Introducere în combinatorică"
- [4] Socaciu Ioan - Tiberiu - "Lucrări de laborator" (manuscris)

Tiberiu Socaciu

```
/* sursa in C */
/* program 1 */
#include  <stdio.h>
#define max 10
int   n,i,nr,x[max];
void prelucreaza(x)
int x[max];
{
  int i;
  nr++;
  for (i=1;i<=n;i++)
    printf("%d;",x[i]);
  printf("\n");
}
int fi(k,x)
int k,x[max];
{
  int i,b;
  b=1;
  for (i=1;i<k;i++) b=b && (x[k]!=x[i]);
return b;
}
void bt(void)
{
  void prelucreaza(int x[max]);
  int fi(int k,int x[max]);
  int  inainte,k;
  k=1;
  x[1]=o;
  while (k>0)
    {
    inainte=0;
    while ((! inainte) && (x[k]<p[k]))
      {
      x[k]++;
      inainte=fi (k,x);
      };
    if (inainte)   if (k==n) prelucreaza(x);
                   else
                   {
                   k++;
                   x[k]=0;
                   }
    else k--;
    }
}
void main(void)
{
  printf("\ndati valoarea lui n:");
  scanf("%d",&n);
  printf("\nvaloarea lui n este:%d\n",n);
  nr=0;
  for (i=1;i<=n;i++)  p[i]=n;
  bt();
  printf("\nnumarul de permutari este:%d",nr);
}
```

```
{sursa in Pascal}
{  program 2  }
type
   vector = array [1..100] of integer;
var
   n,i:integer;
   x,p:vector;
   nr:integer;
procedure prelucreaza(x:vector);
   var
      i:integer;
   begin
      for i: = 1 to n do write(x[i],';');
      writeln;
      nr: = nr + 1
   end;
procedure bt;
   var
      inainte:boolean;
      k  :integer;
   function fi(k:integer;x:vector):boolean;
      var
         i:integer;
         b:boolean;
      begin
         b: = true;
         for i: = 1 to k-1 do
            b: = b and (x[k] < > x[i]);
            fi: = b
         end;
      begin
         k: = 1;
         x[1]: = 0;
         while k > 0 do
            begin
               inainte: = flase;
               while not inainte and (x[k] < > p[k]) do
                  begin
                     x[k]: = x[k] + 1;
                     inainte: = fi(k,x)
                  end;
               if inainte then if k = n then prelucreaza(x)
               else begin
                  k: = k + 1;
                  x[k]: = 0
               end
               else k: = k-1
            end
         end;
      begin
         read(n);
         nr: = 0;
         for i: = 1 to n do p[i]: = n;
         bt;
         writeln(nr)
   end.
```

```
[sursa in Pascal a versiunii recursive)
type
   vector = array [1..100) of integer;
var
   n,i:integer;
   x,p:vector;
   nr:integer;
procedure prelucreaza(x:vector);
   var
      i:integer;
   begin
      for i: = 1 to n do write(x[i],';');
      writeln;
      nr: = nr = 1
   end;
procedure btr(k:integer;var x:vector);
   var
      i:integer;
   function fi(k:integer;x:vector):boolean;
      var
         i:integer;
         b:boolean;
      begin
         b: = true;
         for i: = 1 to k-1 do
            b: = b and (x[k] < > x[i]);
         fi: = b
      end;
   begin
      if fi(k,x) then
      if k = n
         then prelucreaza(x)
         else for i: = 1 to p[k + 1] do
            begin
               x[k + 1]: = 1;
               btr(k + 1,x)
            end
   end;
   begin
      read(n
      nr: = 0;
      for i: = 1 to n do p[i]: = n;
      for i: = 1 to p[1] do
         begin
            x[1]: = i;
            btr(1,x)
         end;
      writeln(nr)
end.
```

Reţele Neuronale

O tehnică de explicare a procesului de luare a deciziilor într-o reţea neuronală.

Unul dintre punctele cheie în care reţelele neuronale diferă faţă de sistemele expert este modul în care acestea îşi structurează intern informaţia conţinută. În sistemele expert tradiţionale, informaţia este ţinută sub formă de reguli. Cînd un sistem expert ajunge la o concluzie, se pot regăsi oricînd în colecţia lui de reguli acele reguli care au fost activate pentru a ajunge la concluzie, şi în ce ordine. În cazul unei reţele neuronale însă, această informaţie nu este la fel de uşor de regăsit, din cauză că reţelele neuronale memorează informaţia sub formă de "accentuări" sau de "întăriturí sinaptice". Accentuările într-o reţea au valori numerice în loc de simbolice, făcînd astfel extrem de dificilă extragerea unei explicaţii simbolice.

Totuşi, este posibil să cunoaştem mai mult despre modul cum afectează diverse intrări ieşirile unei reţele neuronale. Acest lucru este posibil prin aplicarea unei tehnici din domeniul analizei statistice neparametrizate, numită "analiză senzitivă".

Acest articol descrie analiza senzitivă şi cum se aplică ea pentru a descrie "gîndirea" într-o reţea neuronală. Este descris în amănunţime un exemplu însoţit de programe scrise în "C". Exemplul este un procesor complet de reţele neuronale, implementînd algoritmul de propagare înapoi. Acest articol presupune că sînteţi familiarizaţi cu reţelele neuronale şi cu algoritmul de propagare înapoi (vezi articolele începute în numărul 3(6)/91 al revistei "if", despre reţele neuronale).

Ce este analiza senzitivă?

Analiza senzitivă urmăreşte la fiecare intrare modul în care o mică schimbare a ei se răsfrînge în ieşire. Dacă o mică schimbare a unei anumite intrări provoacă mari schimbări unei anumite ieşiri acea intrare poate fi considerată unul dintre factorii cheie ai rezultatelor curente.

Procesul de aplicare a analizei senzitive constă în pornirea cu prima intrare a reţelei şi adăugarea la ea a unei mici cantităţi, recalculînd ieşirile, apoi scăderea unei mici cantităţi din intrarea reţelei şi recalcularea ieşirilor, apoi considerarea diferenţei dintre cele două ieşiri şi raportarea ei la cantitatea totală de schimbare a intrării. De aici rezultă M valori, una pentru fiecare ieşire. Acest proces este repetat pentru fiecare dintre intrările reţelei. Rezultatul este o matrice M x N unde M este numărul ieşirilor iar N numărul intrărilor. Mica modificare a intrării o vom numi "perturbaţie".

În plus, această tehnică de bază este utilizată intern în multe dintre reţelele neuronale pentru a determina care dintre elemntele procesului trebuie modificat.

În particular, ea este utilizat în unele variante ale produsului MADALINE şi în cîteva tehnici de optimizare a reţelelor.

Ca un exemplu despre cum acţionează analiza senzitivă, să presupunem că o reţea neuronală a fost antrenată să determine credibilitatea unui individ. Intrările acestei reţele sînt informaţii de la o aplicaţie de credit incluzînd cîştigurile, cheltuielile, numărul de cărţi de credit, raportul proprietar/chiriaş. Pentru individul

Intrare	Noua ieşire	Modificarea ieşirii
Venit +10%	0,60	+0,30
Venit -10%	0,15	-0,15
Cheltuieli -10%	0,25	-0,05
Cheltuieli + 10%	0,40	+0,10
închiriere → proprietate	0,31	+0,01

Tabelul 1: Efectul schimbării diferitelor variabile de intrare asupra ieşirii curente a unei reţele neuronale antrenată pentru aprobarea creditelor. Ieşirea nominală pentru setul curent de intrări este 0.3.

Scop	Cost în $	Nivel de cumpărare	pentru 10% schimbare în preţ	Senzitivitate
Educaţie	99	1.0	0.00	0.0
Educaţie	150	1.0	-0.20	-2.0
Educaţie	200	0.5	-0.10	-1.0
Educaţie	500	0.3	-0.05	-0.5
Educaţie	1895	0.1	-0.01	-0.1
Aplicaţie	99	0.1	+0.10	+1.0
Aplicaţie	500	0.5	+0.10	+1.0
Aplicaţie	1895	0.8	-0.01	-0.1
Aplicaţie	3000	0.7	-0.01	-0.1

Tabela 2: Schimbări în nivelul de cumpărare. Cît de mult doriţi să vă cheltuiţi banii depinde de scop şi de preţ. Este estimată schimbarea în nivelul de cumpărare. Observaţi că sensibilitatea la o intrare particulară (preţ) depinde de alte intrări (scop). Senzitivitatea este procentul de schimbare în ieşire împărţit la procentul de schimbare în intrare.

curent, să presupunem că ieşirea reţelei este 0.3 (o valoare mai mică decît 0,5 înseamnă respingerea acordării creditului). Tabela 1 arată efectele unei mici schimbări în fiecare intrare asupra ieşirilor. Care dintre intrări va trebui să le schimbaţi puţin pentru a transforma respingerea în acceptare? Cea mai uşoară metodă va fi să cerem individului să-şi crească cîştigul cu 10% rezultînd astfel o acceptare (0.6). Probabil că principala cauză a respingerii este nivelul prea mic de venit al solicitantului. Din tabela 1 rezultă că nu este aproape nici o diferenţă dacă persoana are o casă sau stă în chirie. Pe de altă parte motivul respingerii ar putea fi acela ca individul are cheltuieli prea mari. Dacă aceste cheltuieli ar fi reduse cu 20-30% acest lucru ar putea fi suficient pentru acceptare (ieşirea evaluatorului s-ar ridica peste pragul de 0.5.)

Este foarte important de văzut că modul în care fiecare dintre aceste intrări variabile afectează ieşirea este dependent de valorile lor curente. Pentru un alt individ, cel mai important factor ar putea fi statutul de proprietar sau chiriaş.

În mai multe feluri, aceasta este analog cu modul în care indivizii iau decizii. Cînd e dificil de luat o decizizie între două moduri de derulare a acţiunii, de obicei există un factor particular care este chiar la limita de separaţie (sau mai mulţi). Dacă factorul ar înclina puţin într-o parte sau alta, decizia ar fi clară.

De exemplu, compania mea oferă un pachet soft introductiv pentru 99$ iar un pachet soft sotisficat pentru 1895$. Veţi cheltui 99$ pentru un pachet de învăţare a reţelelor neuronale? Veţi cumpăra fără să gîndiţi prea mult (nivel de cumpărare 1.0). Dar veţi cheltui 1895$? Aproape sigur nu, în afară de cazul în care aveţi o aplicaţie specifică, şi vreţi să fiţi sigur că aveţi cele mai bune scule pentru a avea succesul asigurat (nivel de cumpărare educaţional 0.1, nivel de cumpărare aplicaţional 0.8). Aţi cheltui 200$? Poate (nivel de cumpărare 0.5). Dar 500$? Nu, decît în cazul în care vi-l doriţi foarte mult (nivel de cumpărare 0.3).

Astfel confruntat cu decizia despre cît de mult aţi plăti pentru un pachet soft de învăţare a reţelelor neuronale, cu cît este preţul mai mare, cu atît mai puţin vreţi să îl aveţi. Sub 150$ îl cumpăraţi fără să vă gîndiţi. Peste această sumă aveţi nevoie de motivaţii suplimentare. Dar dacă aveţi o aplicaţie anume în minte, veţi plăti 1895$ bazat pe capacităţile dorite. Dar 3000$? Veţi plăti, bazat pe aceleaţi capacităţi (nivel de cumpărare 0.7). Dar 500$? S-ar putea, dar v-aţi uita mai atent la prezentarea produsului (nivel de cumpărare 0.5). De ce sînteţi mai liniştiţi cumpărînd un produs de 2000-3000$ pentru a rezolva o anume aplicaţie? Pentru că ştiţi din experienţă că sculele bune costă bani, sculele ieftine cerînd de obicei un efort inutil mult mai mare decît banii economisiţi (cu cîteva excepţii notabile).

Priviţi acum cu atenţie tabela 2. Observaţi că un produs educativ este foarte sensibil la preţ cînd preţul urcă peste 150$. Cînd ai nevoie de o sculă pentru dezvoltarea unei aplicaţii, peste o anumită limită de preţ, alţi factori devin mai importanţi decît preţul. Observaţia cheie din această tabelă este că modul în care preţul afectează dorinţa de a cumpăra este dependent de motivul pentru care vreau să cumpăr produsul. Acesta este adevărul analizei senzitive în general. Modificările rezultate în ieşire sînt adevărate numai pentru setul curent de intrări. Pentru că o mică schimbare într-o anumită intrare poate să nu aibă nici un efect la ieşire, nu putem trage concluzia că acea intrare este neimportantă. Combinată cu alte seturi de intrări, o mică schimbare în această intrare poate avea o influenţă puternică asupra ieşirii.

Programul exemplu

Listingul 1 conţine un program pentru o reţea neuronală, proiectat pentru a antrena o reţea cu propagare înapoi şi apoi să o poată explica folosind analiza senzitivă. O rutină principală selectează funcţia dorită şi permite ca reţeaua şi explicaţiile să fie solvate într-un fişier pentru o analiză ulterioară. Pentru că spaţiul este critic reţeaua completă nu este dată împreună cu programul, ci trebuie antrenată. Programul a fost scris sub Turbo C versiunea 2.0 dar trebuie să meargă fără modificări şi sub Microsoft C şi sub Zortech C.

Programul Exemplu

Iată paşii pentru rularea programului exemplu. Următoarea secvenţă va crea network.exe şi va lista tabelele arătate în acest articol.

1. Compilaţi sursa "network.C" (am utilizat Turbo C 2.0 dar veţi putea utiliza Microsoft C sau Zortech C la fel de bine)

C> tca network.C

Dacă aveţi coprocesor de virgulă mobilă, s-ar putea să fie mici diferenţe faţă de acest articol datorită rotunjirilor diferite. Exemplul a fost creat iniţial fără coprocesor.

2. Rulaţi programul Cnetwork.

3. Creaţi o nouă reţea: Ce doriţi să alegeţi? C reţeaua a fost creată.

4. Antrenaţi reţeaua (asta a luat 5 ore pe un Laser 386). Ce doriţi să alegeţi? t se afişează mesaje relative la pasul curent.

5. Salvaţi reţeaua în fişierul "network.net". Ce doriţi să alegeţi?

6. Tipăriţi accentuările reţelei: Ce doriţi să alegeţi? pnetwork.est

7. Analizaţi cum este senzitivă reţeaua la intrări diverse: Ce doriţi să alegeţi? e network.exp

8. Ieşiţi: Ce doriţi să alegeţi? X

Programul este organizat astfel: Rutinele utilitare pentru încărcare, salvare, ştergere, creare şi listare reţea sînt la început. Acestea se pot chema direct din programul principal. Rutina Recall() este utilizată pen-

tru a executa reţeaua o singură dată, pentru determinarea răspunsului său la un anumit set de intrări. Funcţia Learn() execută o singură actualizare a accentuărilor din reţea. Pentru antrenarea accentuărilor se foloseşte metoda standard cu delta şi moment (vezi cartea lui Rummelhart şi McClellant Parallel Distributed Processing, vol 1, cap 8).

Funcţia Train() aplică funcţia Learn() în mod repetat, dîndu-i cîte un exemplu din setul de antrenament, test E (). Pentru a fi siguri că sînt folosite toate exemplele, am folosit o anumită strategie de amestecare, implementată în funcţie NextTestN(), care creează o listă amestecată de exemple, o trimite spre programul se învăţare şi o reamestecă dacă stiva este goală. (vezi textul încasetat pentru detalii de rularre a programului. Au fost luate precauţiile necesare pentru ca programul să fie uşor de compilat sub alte compilatoare C).

Analiza rezultatelor

Reţeaua a fost antrenată pînă cînd eroarea a fost mai mică decît 0.001. Tabelul 4 (a) ne arată ieşirea reţelei peste intervalul intrărilor 1 şi 2. Aceasta corespunde foarte bine antrenamentelor pe intrările test E(). Setul de antrenament a fost conceput astfel încît intrarea 3 să fie aleatoare. Dacă ea ar fi complet aleatoare, accentuările conectate la ea ar trebui să fie toate aproape de zero. Tabelul 3 este o listare a reţelei. Motivaţi că mărimea înregistrărilor asociate cu intrarea 3, subliniate în tabel, sînt aproape toate foarte mici în comparaţie cu accentuările celorlalte intrări la fiecare element de procesare (PE). Aceasta înseamnă că intrarea 3 va avea de obicei un mic efect în ieşirea reţelei. Elementul de procesare 0 este aşa numitul "biaos" şi ieşirea lui este întotdeauna 1.0.

Tabelele 4(b)(c) şi (d) ne arată senzitivitatea ieşirilor la schimbări mici în prima şi a doua variabilă respectiv. Tabelele barează intrările posibile (intrarea 1 pe axa x şi intrarea 2 pe y). Notaţi că ieşirea este cea mai sensibilă cînd o variabilă este mai aproape de margine. Dacă vă gîndiţi un moment, asta are sens. Într-o regiune mai îndepărtată de margine, mici schimbări în intrări vor avea efect mic în ieşire. Doar aproape de margine, cînd ieşirea trece dintr-o stare în alta, o intrare potrivită va avea un efect puternic. Aceasta sugerează unele modificări în program. Anume, dacă o mică schimbare în fiecare intrare, produce puţine, sau nici o schimbare a ieşirilor, să mărim schimbarea pînă efectele sînt semnificative.

În tabela 4(b), care priveşte sensibilitatea ieşirii la schimbări în intrarea 1, observaţi că ieşirea este mai sensibilă cînd intrarea 1 este în domeniul 0.45-0.55 şi intrarea 2 este în domeniul 0.7-1.0. Asta este în locurile unde ieşirea reţelei face tranziţia între 0 şi 1. La fel în tabela 4(c) ieşirea este mai sensibilă la intrarea 2 cînd aceasta este între 0.25 - 0.35 şi 0.65 - 0.8. Aici reţeaua

```
Layer 0 with 3 PEs
 1  1abb:5b12  PE  Output=0.0  Error=0.0  WorkR=0.0  NConns=0
 2  1abb:5b32  PE  Output=0.0  Error=0.0  WorkR=0.0  NConns=0
 3  1abb:5b52  PE  Output=0.0  Error=0.0  WorkR=0.0  NConns=0

Layer 1 with 5 PEs
 4  1abb:5b72  PE  Output=0.0  Error=0.0  WorkR=0.0  NConns=4
   Src=0  1abb:00d6  Weight=   1.591  Delta Wt=  0.0
   Src=1  1abb:5b12  Weight=   3.830  Delta Wt=  0.0
   Src=2  1abb:5b32  Weight= -16.104  Delta Wt=  0.0
   Src=3  1abb:5b52  Weight=   0.486  Delta Wt=  0.0
 5  1abb:5bba  PE  Output=0.0  Error=0.0  Workr=0.0  NConns=4
   Src=0  1abb:00d6  Weight=  14.082  Delta Wt=  0.0
   Src=0  1abb:5b12  Weight= -15.446  Delta Wt=  0.0
   Src=0  1abb:5b32  Weight= -16.720  Delta Wt=  0.0
   Src=0  1abb:5b52  Weight=   2.472  Delta Wt=  0.0
 6  1abb:5c02  PE  Output=0.0  Error=0.0  Workr=0.0  NConns=4
   Src=0  1abb:00d6  Weight=   3.883  Delta Wt=  0.0
   Src=0  1abb:5b12  Weight=   2.332  Delta Wt=  0.0
   Src=0  1abb:5b32  Weight= -20.745  Delta Wt=  0.0
   Src=0  1abb:5b52  Weight=   1.366  Delta Wt=  0.0
 7  1abb:5c4a  PE  Output=0.0  Error=0.0  Workr=0.0  NConns=4
   Src=0  1abb:00d6  Weight=  -1.377  Delta Wt=  0.0
   Src=0  1abb:5b12  Weight=  -3.519  Delta Wt=  0.0
   Src=0  1abb:5b32  Weight=  -3.751  Delta Wt=  0.0
   Src=0  1abb:5b52  Weight=   0.545  Delta Wt=  0.0
 8  1abb:5c92  PE  Output=0.0  Error=0.0  Workr=0.0  NConns=4
   Src=0  1abb:00d6  Weight=   4.869  Delta Wt=  0.0
   Src=0  1abb:5b12  Weight= -10.058  Delta Wt=  0.0
   Src=0  1abb:5b32  Weight=  -8.766  Delta Wt=  0.0
   Src=0  1abb:5b52  Weight=   3.108  Delta Wt=  0.0

Layer 2 with 2 PEs
 9  1abb:5cda  PE  Output=0.0  Error=0.0  Workr=0.0  NConns=9
   Src=0  1abb:00d6  Weight= -20.434  Delta Wt=  0.0
   Src=1  1abb:5b12  Weight=  46.298  Delta Wt=  0.0
   Src=2  1abb:5b32  Weight=  -2.895  Delta Wt=  0.0
   Src=3  1abb:5b52  Weight=  -0.492  Delta Wt=  0.0
   Src=4  1abb:5b72  Weight= -21.351  Delta Wt=  0.0
   Src=5  1abb:5bba  Weight=  20.866  Delta Wt=  0.0
   Src=6  1abb:5c02  Weight= -26.287  Delta Wt=  0.0
   Src=7  1abb:5c4a  Weight=   1.304  Delta Wt=  0.0
   Src=8  1abb:5c92  Weight=   8.782  Delta Wt=  0.0
10  1abb:5d54  PE  Output=0.0  Error=0.0  Workr=0.0  NConns=9
   Src=0  1abb:00d6  Weight=  20.434  Delta Wt=  0.0
   Src=1  1abb:5b12  Weight= -46.298  Delta Wt=  0.0
   Src=2  1abb:5b32  Weight=   2.895  Delta Wt=  0.0
   Src=3  1abb:5b52  Weight=   0.492  Delta Wt=  0.0
   Src=4  1abb:5b72  Weight=  21.351  Delta Wt=  0.0
   Src=5  1abb:5bba  Weight= -20.866  Delta Wt=  0.0
   Src=6  1abb:5c02  Weight=  26.287  Delta Wt=  0.0
   Src=7  1abb:5c4a  Weight=  -1.292  Delta Wt=  0.0
   Src=8  1abb:5c92  Weight=  -8.783  Delta Wt=  0.0
```

Tabela3: Accentuările în reţeaua antrenată.Rîndurile subliniate sînt accentuările corespunzătoare intrării 3. Intrarea 3 a fost proiectată mai mult sau mai puţin aleatoare. Ca un rezultat, accentuările asociate cu ea sînt în general mici relativ la celelalte accentuări pentru elementele de procesare (PE).

face tranziţia între 0 şi 1 respectiv 1 şi 0, de-a lungul axei pentru intrarea 2.

Uitîndu-ne la tabelele 4(b)(c) şi (d) observăm că ieşirea este întotdeauna sensibilă la intrare cînd la intrare se întîmplă o tranziţie. De altfel exemplul are margini foarte clare şi o singură variabilă se poate schimba la fiecare dintre ele. Asta vă poate face să suspectaţi un anumit fel de probleme. Problema apare datorită modului în care o reţea cu propagare înapoi îşi crează ieşirea. Ea ia o serie de ridicături şi le potriveşte

împreună pentru a construi o anumită stare a intrărilor şi ieşirilor.

Priviţi din nou spre exemplul despre acceptarea creditelor. Schimbările în variabilele de intrare au fost procentaje din intrări neprelucrate. În funcţie de cum au fost modificate aceste variabile de intrare, asta poate însemna o schimbare mai mare sau mai mică în una sau mai multe dintre ieşirile reţelei. În astfel de situaţii e important să variem domeniul de intrare şi să măsurăm schimbările în ieşire.

(a)

1.00	-	-	-	-	-	-	-	-	-	-	0.4	0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	-	-	-	-	-	-	-	-	-	-	0.4	0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
0.90	-	-	-	-	-	-	-	-	-	-	0.5	0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	-	-	-	-	-	-	-	-	-	-	0.5	0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
0.80	0.2	-	-	-	-	-	-	-	-	0.1	0.6	0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	0.7	0.7	0.5	0.2	-	-	-	-	-	0.2	0.6	0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
0.70	0.9	1.0	1.0	0.9	0.7	0.2	-	-	-	0.3	0.7	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	1.0	1.0	1.0	1.0	1.0	1.0	0.8	0.5	0.4	0.5	0.8	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
0.60	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.9	0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
0.50	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
0.40	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.8	0.9	0.9	0.9	1.0
0.30	0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.7	-	-	-	-	-	-	-
	-	0.1	0.3	0.5	0.6	0.7	0.7	0.7	0.6	0.4	0.2	0.1	-	-	-	-	-	-	-	-	-
0.20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0.10	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0.00	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	0.00		0.1		0.2		0.3		0.4		0.5		0.6		0.7		0.8		0.9		1.0

(b)

1.00	-	-	-	-	-	-	-	-	0.3	2.7	10.9	5.3	0.7	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	0.4	3.1	11.1	4.8	0.6	-	-	-	-	-	-	-	-
0.90	-	-	-	-	-	-	-	-	0.4	3.5	11.3	4.2	0.5	-	-	-	-	-	-	-	-
	-0.2	-	-	-	-	-	-	-	0.5	4.1	11.2	3.7	0.4	-	-	-	-	-	-	-	-
0.80	-1.3	-1.9	-0.5	-	-	-	-	-	0.7	4.7	10.9	3.2	0.4	-	-	-	-	-	-	-	-
	1.3	-1.5	-6.5	-5.5	-0.9	-	-	0.2	1.0	5.6	10.1	2.7	0.3	-	-	-	-	-	-	-	-
0.70	1.0	0.3	-0.2	-1.5	-7.9	-7.6	-1.3	0.1	1.6	6.6	8.4	2.1	0.3	-	-	-	-	-	-	-	-
	0.4	0.1	-	-	-0.1	-0.9	-5.7	-5.8	0.4	5.4	5.2	1.4	0.2	-	-	-	-	-	-	-	-
0.60	0.2	-	-	-	-	-	-	-.05	-1.6	-	1.4	0.7	0.1	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-0.1	-	0.1	-	-	-	-	-	-	-	-	-
0.50	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0.40	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-0.1	-	-	-	-	-	-	-	-	-	-	-	-	-0.2	-1.3	-1.5	-0.2	0.7	0.8	0.6	0.4
0.30	1.7	0.4	-	-	-	-	-	-	-	-	-	-	-1.1	-15.8	-3.4	-0.1	-	-	-	-	-
	1.0	2.2	3.5	3.6	2.4	1.0	-0.2	-1.5	-2.8	-3.4	-3.1	-2.2	-1.0	-0.2	-	-	-	-	-	-	-
0.20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0.10	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0.00	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	0.0		0.1		0.2		0.3		0.4		0.5		0.6		0.7		0.8		0.9		1.0

(Tabela 4 se continuă pe pagina următoare)

Tabela 4: (a) Ieşirea reţelei ca funcţie de intrările 1 şi 2 cu intrarea 3 ţinută constant la 0.5. Observaţi că reţeaua a învăţat configuraţia dorită foarte bine. (b) Rezultatul aplicării analizei senzitive pentru intrarea 1 peste domeniul valorilor pentru intrarea 1 şi 2. Ieşirea este sensibilă la tranziţie cînd intrarea 1 merge de la 0.45 la 0.55 (şi ieşirea de la 0 la 1). (c) Rezultatul aplicării analizei senzitive pentru intrarea 2 peste domeniul valorilor intrărilor 1 şi 2. Ieşirea este foarte senzitivă la două tranziţii. Prima cînd intarea 2 este între 0.25 şi 0.35 şi a doua cînd intrarea 2 este între 0.65 şi 0.75. În primul caz senzitivitatea este negativă cînd ieşirea face o tranziţie între 0 şi 1. În al doilea caz (pentru intrarea 1 mai mică decît 0.5)senzitivitatea este negativă cînd ieşirea face o tranziţie în direcţia negativă de la 1 la 0. (d) Deşi ieşirea este senzitivă la intrările 3, valorile sînt întotdeauna mai mici. Acesta era de aşteptat de la o intrare aleatoare.

Pentru dezvoltări ulterioare

Programul conţine mai multe posibilităţi de dezvoltare ulterioară. Cum am descris mai înainte, făcînd

perturbaţia mai mare, se obţin efecte interesante. La un anumit grad, se poate găsi un nivel de încredere în funcţie de cît de mare este perturbaţia. O perturbaţie mică din care rezultă o senzitivitate mare înseamnă o concluzie mai puţin sigură decît nici o senzitivitate. În acest program, efectele unei perturbaţii pozitive şi negative au fost mediate, dar ele s-ar putea raporta şi separat.

Dacă setăm o eroare mai mare în rezultatele antrenării, rezultă margini mai neclare. Asta tinde să facă reţeaua mai puţin sensibilă la oricare intrări. Se crează astfel regiuni în care propagarea înapoi nu mai funcţionează corect.

Setul de date care a fost luat în program a fost special construit pentru a arăta cum funcţionează analiza senzitivă. Încercaţi să construiţi şi alte seturi de date, care au margini neclare sau neregulate.

(Traducere şi adaptare: mat. Eugen Rotariu)

(c)

1.00	-	-	-	-	-	-	-	-	-	-0.2	-0.7	-0.3	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-0.2	-0.8	-0.3	-	-	-	-	-	-	-	-	-	-
0.90	-	-	-	-	-	-	-	-	-	-0.3	-0.8	-0.3	-	-	-	-	-	-	-	-	-	-
	-0.5	-	-	-	-	-	-	-	-	-0.4	-0.9	-0.3	-	-	-	-	-	-	-	-	-	-
0.80	-82	-5.7	-1.3	-0.2	-	-	-	-	-0.1	-0.6	-1.0	-0.2	-	-	-	-	-	-	-	-	-	-
	-8.6	-11.7	-18.5	12.8	-2.2	-0.4	-0.2	-0.2	-0.4	-1.2	-1.3	-0.2	-	-	-	-	-	-	-	-	-	-
0.70	-1.9	-1.5	-1.9	-5.1	-18.1	-17.4	-4.6	-1.9	-2.0	-3.0	-1.9	-0.3	-	-	-	-	-	-	-	-	-	-
	-0.4	-0.2	-0.1	-0.2	-0.4	-2.5	13.5	-18.3	-11.3	-7.1	-2.4	-0.3	-	-	-	-	-	-	-	-	-	-
0.60	-0.1	-	-	-	-	-	-0.2	-1.4	-5.2	-5.5	-2.0	-0.3	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-0.1	-0.5	-0.6	-0.2	-	-	-	-	-	-	-	-	-	-
0.50	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0.40	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	0.1	-	-	-	-	-	-	-	-	-	-	-	-	0.3	4.5	16.7	24.5	25.8	23.3	19.0	14.1	
0.30	5.1	1.4	0.4	0.1	-	-	-	-	-	-	0.1	0.4	3.3	25.6	7.6	0.7	0.2	-	-	0.1	0.2	
	7.7	18.3	29.4	34.8	34.9	33.5	33.0	34.3	36.1	35.7	30.6	19.1	5.6	0.5	-	-	-	-	-	-	-	
0.20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
0.10	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
0.00	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
	0.00		0.1		0.2		0.3		0.4		0.5		0.6		0.7		0.8		0.9		1.0	

(d)

1.00	-	-	-	-	-	-	-	-	-	-	-0.1	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-0.1	-	-	-	-	-	-	-	-	-	-	-
0.90	-	-	-	-	-	-	-	-	-	-	-0.1	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0.80	1.6	1.6	0.2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	1.9	2.3	3.4	2.0	0.3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0.70	0.5	0.3	0.4	0.8	3.0	2.7	0.6	0.3	0.2	0.3	0.1	-	-	-	-	-	-	-	-	-	-	-
	0.1	-	-	-	-	0.4	2.0	2.8	1.6	0.9	0.3	-	-	-	-	-	-	-	-	-	-	-
0.60	-	-	-	-	-	-	-	0.2	0.7	0.8	0.3	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0.50	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0.40	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-0.5	-1.0	-1.2	-1.0	-0.7	-0.5	
0.30	-0.3	0.4	-	-	-	-	-	-	-	-	-	-	-	0.5	-	-	-	-	-	-	-	
	-0.4	-1.0	-1.8	-2.3	-2.0	-1.6	-1.3	-1.2	-1.1	-1.0	-0.7	-0.3	-	-	-	-	-	-	-	-	-	
0.20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
0.10	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
0.00	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
	0.0		0.1		0.2		0.3		0.4		0.5		0.6		0.7		0.8		0.9		1.0	

```
#include   <stdio.h>   /* pentru lucru cu fisiere */
#include   <stdlib.h>   /* malloc, rand, RAND_MAX, etc. */
#include   <math.h>   /* exp(), sqrt() */
#include   <dos.h>   /* FP_SEG(), FP_OFF() */

#define MAX_LAYER 4  /* numarul maxim de nivele */
#define MAX_PES 50 /* numarul maxim de elemente de procesare */

#define WORD(x) (((short *)(&x))[0])
#define MAX(x,y) ((x)(y)?(x):(y))
typedef float WREAL;  /* real pentru lucru */

/* structura de conectare */
typedef struct _conn {
  struct      _pe  *SourceP;     /* pointer sursa */
  WREAL       WtValR;   /* valoarea ingrosarii */
  WREAL       DWtValR;   /* delta ingrosare */
} CONN;

/* element de procesare */
typedef struct _pe {
  struct      pe  *NextPEP;  /* urmatorul PE pe nivel */
  WREAL       OutputR;   /* iesirea PE-ului */
  WREAL       ErrorR;   /* zona de lucru pentru eroare */
  WREAL       WorkR;   /* zona de lucru pentru functia de
explicatie */
  int *PEIndexN;  /* indexul PE-ului in PEIndexP */
  int MaxConnsN; /* numarul maxim de conexiuni */
  int NConnsN;   /* numarul de conexiuni utilizate */
  CONN  ConnsS[1]; /* lista conexiunilor acestui PE */
} PE;

PE *LayerTP[MAX_LAYER+1] = {0}; /* adresele PE-urilor de pe
nivel */
int LayerNI[MAX_LAYER+1] = {0}; /* numarul PE-urilor de pe
fiecare nivel */

PE *PEIndexP[MAX_PES] = {0};   /* adresele PE-urilor */
int NextPEXN = {0};          /* indexul urmatorului PE liber */
PE PEBias = {0, 1.0, 0.0, 0, };

/***** RRandR() - calculeaza o valoare aleatoare intr-un interval */
double RRandR( vR )
double vR;       /* marimea intervalului */
{
 double rvR;   /* valoarea returnata */

  /* valoarea aleatoare intre 0 si 1 */
  rvR = ((double)rand()) / (double)RAND_MAX;
  /* transformata intre -vR si vR */
  rvR = vR * (rvR + rvR - 1.0);
  return( rvR );
  }

/***** AllocPE() - aloca dinamic un PE */
PE *AllocPE( peXN, MaxConnsN )
int peXN;        /* indexul PE-ului (0 = automat) */
int MaxConnsN;    /* numarul maxim de conexiuni */
{
  PE *peP;
  int AlcSize;

  if ( NextPEXN = = 0 ) {
  PEIndexP[0] = &PEBias;
  NextPEXN + +;
  }
  if ( peXN = = 0 )
    peXN = NextPEXN + +;
  else if ( peXN > = NextPEXN )
    NextPEXN = peXN+1;

  if ( peXN < 0 || MAX_PES  < = peXN ) {
  printf( "Valoare ilegala a indexului PE-ului de alocat: %d\n",
    peXN );
  exit( 1 );
  }
  if ( PEIndexP[peXN] != (PE *) 0 ) {
    printf( "PE-ul numarul %d este deja utilizat\n", peXN );
    exit( 1 );
  }
  AlcSize = sizeof( PE ) + MaxConnsN * sizeof( CONN );
  peP = (PE *) malloc( AlcSize );
  if ( peP = = (PE *) 0 ) {
    printf( "Nu pot aloca %d octeti pentru PE-ul numarul %d\n",
      AlcSize , peXN );
    exit( 1 );
  }
  memset( (char *) peP, 0, AlcSize );
  peP->MaxConnsN = MaxConnsN+1; /* numarul maxim
          de conexiuni */
  peP->PEIndexN = peXN;       /* index folosit pentru incarca-
re/salvare */
  PEIndexP[peXN] = peP; /* index pentru mai tirziu */
  return( peP );
}

/***** AllocLayer() - Aloca dinamic PE-urile de pe un nivel */
int AllocLayer( LayN, NPEsN, NConnPPEN )
int LayN;       /* numarul nivelului */
int NPEsN;       /* numarul de PE-uri de pe nivel */
int NConnPPEN;    /* numarul de conexiuni pentru fiecare PE */
{
  PE *peP;
  PE *apeP;
  int wxN;

  /* verificare de siguranta */
  if (LayN  < 0 || sizeof( LayerTP)/sizeof( LayerTP[0] ) < = LayN)
  {
    printf( "Numar de nivel incorect (%d)\n", LayN );
    exit( 1 );
  }
  /* aloca PE-urile din nivel si le leaga impreuna */
  LayerNI[LayN] = NPEsN;
  for( wxN = 0; wxN  NPEsN; wxN+ +, apeP = peP ) {
  peP = AllocPE( 0, NConnPPEN+1 );   /* aloca urmatorul PE */
  if ( LayerTP[LayN] = = (PE *) 0 )
    LayerTP[LayN] = peP;
  else
    apeP->NextPEP = peP;
  }
  return( 0 );
}

/***** FreeNet() - elibereaza memoria ocupata de retea *****/
int FreeNet ()
{
  int wxN;
  char *P;

  for( wxN = 1; wxN < MAX_PES; wxN+ + ) {
  if ( (P = (char *) PEIndexP[wxN]) != (char *) 0 )
    free( P );
    PEIndexP[wxN] = (PE *) 0;
  }
  NextPEXN = 0;
  for( wxN = 0; wxN <MAX_LAYER; wxN+ + ) {
  LayerTP[wxN] = (PE *) 0;
  LayerNI[wxN] = 0;
  }
  return( 0 );
}

/***** SaveNet() - salveaza reteaua intr-un fisier */
```

```
int SaveNet( fnP )
char *fnP;        /* numele fisierului in care se listeaza */
{
  int wxN;
  FILE *fP;
  PE  *peP;
  CONN *cP;
  int  ConnXN;

  if ( NextPEXN < = 1 )
    return( 0 );
  if ( ( fP = fopen( fnP, "w" )) = = (FILE *) 0 ) {
    printf( "Nu pot deschide fisierul de iesire \n", fnP );
    return( -1 );
  }
  /* salveaza toate PE-urile */
  for( wxN = 1; wxN < NextPEXN; wxN+ + ) {
  peP = PEIndexP[wxN];
    if ( peP = = (PE *) 0 ) {
      fprintf( fP, "%d 0 PE\n", wxN );
      continue;
    }
  fprintf( fP, "%d %d  PE\n", wxN, peP-NConnsN );
    for( ConnXN = 0; ConnXN <peP->NConnsN; ConnXN+ + ) {
      cP = &peP->ConnsS[ConnXN];
      fprintf( fP, "%d %.6f %.6f\n",
       cP->SourceP->PEIndexN, cP->WtValR, cP->DWtValR );
    }
  }
  fprintf( fP, "%d %d Sfirsitul PE-urilor\n", -1, 0 );
  /* salveaza modul in care sint legate PE-urile */
  for( wxN = 0; wxN < MAX_LAYER; wxN+ + ) {
  if ( ( peP = LayerTP[wxN] ) = = (PE *) 0 )
      continue;
    fprintf( fP, "%d NIVEL\n", wxN );
    do {
      fprintf( fP, "%d\n", peP->PEIndexN );
      peP = peP->NextPEP;
    } while ( peP != (PE *) 0 );
    fprintf( fP, "-1 Sfirsitul Nivelului\n" ); /* sfirsitul nivelului */
  }
  fprintf( fP, "-1\n" ); /* Sfirsitul tuturor nivelelor */
  fclose( fP );
  return( 0 );
}

/***** LoadNet() - incarca o retea dintr-un fisier */
int LoadNet( fnP )
char *fnP;        /* numele fisierului de unde se incarca */
{
  int wxN;
  FILE *fP;
  PE *peP;
  PE *lpeP;
  int  LayN;
  CONN  *cP;
  int ConnXN;
  int PEN;
  int PENConnsN;
  int StateN;
  WREAL WtR, DWtR;
  char BufC[80];

  fP = (FILE *) 0;
  if ( ( fP = fopen( fnP , "r" ) ) = = (FILE *) 0 ) {
    printf( "Nu pot deschide fisierul de intrare \n", fnP );
    return( -1 );
  }
  FreeNet();     /* renunta la reteaua deja existenta */
  StateN = 0;
  while( fgets( BufC, sizeof( BufC ) - 1, fP ) != (char *) 0 ) {
    switch ( StateN ) {
      case 0:   /* PE-uri */
        sscanf( BufC, "%d %d", &PEN, &PENConnsN );
        if ( PEN < 0 ) {
          StateN = 2;
          break;
        }
        peP = AllocPE( PEN, PENConnsN );
        cP = &peP->ConnsS[0];
        ConnXN = PENConnsN;
        if ( ConnXN > 0 ) {
          StateN = 1;
        }
        break;
      case 1:   /* Conexiuni PE-uri */
        sscanf( BufC, "%d %f %f", &PEN, &WtR, &DWtR );
        WORD( cP->SourceP ) = PEN;
        cP->WtValR = WtR;
        cP->DWtValR = DWtR;
        cP+ +;
        peP->NConnsN+ +;
        if ( -- ConnXN < = 0 )
          StateN = 0;
        break;
      case 2:   /* Date despre nivele */
        sscanf( BufC, "%d", &LayN );
        StateN = 3;
        if ( LayN < 0 )
          goto Done;
        lpeP = (PE *)&LayerTP[LayN];
        break;
      case 3:   /* Nivelele */
        sscanf( BufC, "%d", &PEN );
        if ( PEN < 0 ) {
          StateN = 2;
          break;
        }
        LayerNI[LayN]+ +;      /* numarul de PE-uri */
        peP = PEIndexP[PEN];   /* adresa PE-ului */
        lpeP->NextPEP = peP;
        lpeP = peP;
        break;
    }
  }
Done:
  /* converteste indexurile tuturor PE-urilor in pointeri */
  for( wxN = 1; wxN < MAX_PES; wxN+ + ) {
    if ( ( peP = PEIndexP[wxN]) = = (PE *) 0 )
      continue;
    for ( ConnXN = peP->NConnsN, cP = &peP->ConnsS[0];
      --ConnXN > = 0; cP+ + ) {
      cP->SourceP = PEIndexP[ WORD(cP->SourceP) ];
    }
  }
  if ( fP )
    fclose( fP );
  return( 0 );
ErrExit:
  if ( fP )
    fclose ( fP );
  FreeNet();
  return( -1 );
}

/***** PrintNet() - Tipareste reteaua */
int PrintNet( fnP )
char *fnP;    /* numele fisierului */
{
   FILE *fP;
  PE *dpeP;
  int layerXN;
  CONN *cP;
  int ConnXN;
```

```
  if ( *fnP == '\0' ) {
    fP = stdout;
  }
  else {
    if ( ( fP = fopen( fnP, "a" ) ) == (FILE *) 0 ) {
      printf ( "Nu pot deschide fisierul de iesire <%s>\n", fnP );
      return( -1 );
    }
  }
  for( layerXN = 0; ( dpeP = LayerTP[layerXN] ) != (PE *) 0; layerXN++ ) {
    fprintf( fP, "\nNivelul %d cu %d PE-uri\n",
      layerXN, LayerNI[layerXN] );
    for( ; dpeP != (PE *) 0; dpeP = dpeP->NextPEP ) {
      fprintf( fP,
        " %2d %04x::%04x PE Iesire = %6.3f Eroare = %6.3f
        Variabila de lucru = %6.3f Numar conexiuni = %d\n",
        dpeP->PEIndexN, FP_SEG(dpeP), FP_OFF(dpeP),
        dpeP->OutputR, dpeP->ErrorR, dpeP->WorkR,
        dpeP->NConnsN );
      for(ConnXN=0;ConnXN<dpeP->NConnsN;ConnXN++) {
        cP = &dpeP->ConnsS[ConnXN];
        fprintf( fP,
          "  Sursa  %2d %04x::%04x Ingrosare = %7.3f
          Delta Ingr. =  6.3f\n",
          cP->SourceP->PEIndexN, FP_SEG(cP->SourceP),
          FP_OFF(cP->SourceP), cP->WtValR, cP->DWtValR );
      }
    }
  }
  if ( fP != stdout )
    fclose( fP );
  return( 0 );
}

/***** FullyConn() - conecteaza total un nivel sursa la o destinatie */
int FullyConn( DLayN, SLayN, RangeR )
int DLayN;    /* nivelul destinatie */
int SLayN;    /* nivelul sursa */
double RangeR; /* intervalul de marime */
{
  CONN *cP;
  PE *speP;
  PE *dpeP;

  /* pentru toate PE-urile de pe nivelul destinatie */
  for( dpeP = LayerTP[DLayN]; dpeP != (PE *) 0;
       dpeP = dpeP->NextPEP ) {
    cP = &dpeP->ConnsS[dpeP->NConnsN];
    if ( dpeP->NConnsN == 0 ) {
      cP->SourceP = &PEBias;
      cP->WtValR = RRandR( RangeR );
      cP->DWtValR = 0.0;
      cP++;
      dpeP->NConnsN++;
    }
    /* pentru toate PE-urile de pe nivelul sursa */
    for( speP = LayerTP[SLayN]; speP != (PE *) 0;
         speP = speP->NextPEP ) {
      cP->SourceP = speP;
      cP->WtValR = RRandR( RangeR );
      cP->DWtValR = 0.0;
      cP++;
      dpeP->NConnsN++;
    }
  }
  return( 0 );
}

/***** BuildNet() - creaza toate structurile necesare pentru retea */
int BuildNet( NInpN, NHid1N, NHid2N, NOutN, ConnPrevF )
int NInpN;    /* numarul de PE-uri de intrare */
int NHid1N;   /* numarul de PE-uri pe nivelul intermediar 1 */
int NHid2N;   /* numarul de PE-uri pe nivelul intermediar 2 */
int NOutN;    /* numarul de PE-uri de iesire */
int ConnPrevF; /* 1 = conecteaza la toate nivelele anterioare */
{
  int ReqdPEsN;
  int LayerXN;
  int SLayN, DLayN;

  if ( NInpN <= 0 || NOutN <= 0 )
    return( -1 );
  FreeNet();
  LayerXN = 0;
  AllocLayer( LayerXN, NInpN, 0 );
  if ( NHid1N > 0 ) {
    LayerXN++;
    AllocLayer( LayerXN, NHid1N, NInpN );
    if ( NHid2N > 0 ) {
      LayerXN++;
      AllocLayer( LayerXN, NHid2N, NHid1N + ( ConnPrevF ?
NInpN : 0 ) );
    }
  }
  LayerXN++;
  AllocLayer( LayerXN, NOutN, ConnPrevF ? ( NInpN + NHid1N
+ NHid2N ) : NHid2N );
  for( DLayN = 1; LayerTP[DLayN] != (PE *) 0; DLayN++ ) {
    for( SLayN = ConnPrevF ? 0 : ( DLayN - 1 ); SLayN < DLayN;
         SLayN++ )
      FullyConn( DLayN, SLayN, 0.2 );
  }
  return( 0 );
}

/***** Recall() - executa un pas in retea */
int Recall( ivRP, ovRP )
WREAL *ivRP;      /* vectorul de intrare */
WREAL *ovRP;      /* vectorul de iesire */
{
  int DLayN;
  PE *peP;
  CONN *cP;
  int ConnC;
  double SumR;

  for( DLayN = 0; ( peP = LayerTP[DLayN] ) != (PE *) 0;
       DLayN++ ) {
    for( ; peP != (PE *) 0; peP = peP->NextPEP ) {
      if ( DLayN == 0 ) {
        peP->OutputR = ivRP[0];
        peP->ErrorR = 0.0;
        ivRP++;
      }
      else {
        ConnC = peP->NConnsN;
        cP = &peP->ConnsS[0];
        SumR = 0.0;
        for( ; --ConnC >= 0; cP++ )
          SumR += cP->SourceP->OutputR * cP->WtValR;
        peP->OutputR = 1.0 / ( 1.0 + exp( - SumR ) );
        peP->ErrorR = 0.0;
      }
      if ( LayerTP[DLayN+1] == (PE *) 0 ) {
        ovRP[0] = peP->OutputR;
        ovRP++;
      }
    }
  }
  return( 0 );
}

/***** Learn() - un pas elementar in procesul de invatare */
```

```
double Learn( ivRP, ovRP, doRP, LearnR, MomR )
WREAL *ivRP;      /* vector de intrare */
WREAL *ovRP;      /* vector de iesire */
WREAL *doRP;      /* vector dorit */
double LearnR;    /* rata de invatare */
double MomR;      /* momentul */
{
  double ErrorR = 0.0;
  double LErrorR;
  PE *speP;
  int DLayN;
  PE *peP;
  CONN *cP;
  int ConnC;

  Recall( ivRP, ovRP );

  for( DLayN = 0; ( LayerTP[DLayN+1] ) != (PE *) 0; )
    DLayN++;
  /* calculeaza eroarea, propagare inapoi, actualizare ingrosari */
  for( ; DLayN > 0; DLayN-- ) {
    for( peP = LayerTP[DLayN]; peP != (PE *) 0;
         peP = peP->NextPEP ) {
      if ( LayerTP[DLayN+1] == (PE *) 0 ) {
        peP->ErrorR = ( doRP[0] - peP->OutputR );
        ErrorR += ( peP->ErrorR * peP->ErrorR );
        doRP++;
      }
      peP->ErrorR *= peP->OutputR * ( 1.0 - peP->OutputR );
      ConnC = peP->NConnsN;
      cP = &peP->ConnsS[0];
      LErrorR = peP->ErrorR;
      for( ; --ConnC >= 0; cP++ ) {
        speP = cP->SourceP;
       /* propaga eroarea */
        speP->ErrorR += LErrorR * cP->WtValR;
        cP->DWtValR = LearnR * LErrorR * speP->OutputR +
            MomR * cP->DWtValR;
        cP->WtValR += cP->DWtValR;  /* actualizeaza ingrosa-
rile */
      }
    }
  }
  return( ErrorR );
}

/***** Explain() - calculeaza iesirile corespunzatoare
modificarilor in intrare */
int Explain( ivRP, ovRP, evRP, DitherR )
WREAL *ivRP;      /* vector de intrare */
WREAL *ovRP;      /* vector de iesire */
WREAL *evRP;      /* vector de explicatie */
double DitherR;   /* perturbatie */
{
  PE *speP;
  int speXN;
  PE *dpeP;
  int dpeXN;
  int OutLXN;

  for( OutLXN = 0; LayerTP[OutLXN+1] != (PE *) 0; )
    OutLXN++;
  Recall( ivRP, ovRP );
  for( dpeP = LayerTP[OutLXN]; dpeP != (PE *) 0;
       dpeP = dpeP->NextPEP )
    dpeP->WorkR = dpeP->OutputR;
  for( speXN = 0, speP = LayerTP[0]; speP != (PE *) 0;
       speXN++, speP = speP->NextPEP ) {
    /* perturbatie positiva */
    ivRP[speXN] += DitherR;
    Recall( ivRP, ovRP );
    for( dpeXN = 0, dpeP = LayerTP[OutLXN]; dpeP != (PE *) 0;
         dpeXN++, dpeP = dpeP->NextPEP )
      evRP[dpeXN] = 0.5 * ( dpeP->OutputR - >dpeP->WorkR ) /
          DitherR;
    /* perturbatie negativa */
    ivRP[speXN] -= ( DitherR + DitherR );
    Recall( ivRP, ovRP );
    for( dpeXN = 0, dpeP = LayerTP[OutLXN]; dpeP != (PE *) 0;
         dpeXN++, dpeP = dpeP->NextPEP )
      evRP[dpeXN] -= 0.5 * ( dpeP->OutputR - dpeP->WorkR ) /
          DitherR;
    evRP += dpeXN;
    ivRP[speXN] += DitherR;
  }
  return( 0 );
}

/***** date pentru antrenarea retelei:
        +-------------------------------+  1.0
   I    |  zero  |                      |
   N    |        |         one          |
   T    +--------+                      |  0.7
   R    |                               |
   A    |                               |
   R    +-------------------------------+  0.3
   E    |                               |
   A    |              zero             |
   2    +-------------------------------+  0.0
       0.0             0.5             1.0
                INTRAREA 1

Intrarea 3 este "zgomot".
*/

typedef struct _example {
  WREAL InVecR[3];      /* vector de intrare */
  WREAL DoVecR[2];      /* vector de iesire dorit */
} EXAMPLE;

#define NTEST ( sizeof( testE ) / sizeof( testE[0] ) )

EXAMPLE testE[] = {
  { 0.0, 0.0, 0.0,  0.0, 1.0 },
  { 0.0, 0.2, 0.6,  0.0, 1.0 },
  { 0.0, 0.4, 0.1,  1.0, 0.0 },
  { 0.0, 0.6, 0.7,  1.0, 0.0 },
  { 0.0, 0.8, 0.2,  0.0, 1.0 },
  { 0.0, 1.0, 0.8,  0.0, 1.0 },

  { 0.2, 0.0, 0.9,  0.0, 1.0 },
  { 0.2, 0.2, 0.3,  0.0, 1.0 },
  { 0.2, 0.4, 0.8,  1.0, 0.0 },
  { 0.2, 0.6, 0.2,  1.0, 0.0 },
  { 0.2, 0.8, 0.7,  0.0, 1.0 },
  { 0.2, 1.0, 0.1,  0.0, 1.0 },

  { 0.4, 0.0, 0.0,  0.0, 1.0 },
  { 0.4, 0.2, 0.6,  0.0, 1.0 },
  { 0.4, 0.4, 0.1,  1.0, 0.0 },
  { 0.4, 0.6, 0.7,  1.0, 0.0 },
  { 0.4, 0.8, 0.2,  0.0, 1.0 },
  { 0.4, 1.0, 0.8,  0.0, 1.0 },

  { 0.6, 0.0, 0.9,  0.0, 1.0 },
  { 0.6, 0.2, 0.3,  0.0, 1.0 },
  { 0.6, 0.4, 0.8,  1.0, 0.0 },
  { 0.6, 0.6, 0.2,  1.0, 0.0 },
  { 0.6, 0.8, 0.7,  1.0, 0.0 },
  { 0.6, 1.0, 0.1,  1.0, 0.0 },

  { 0.8, 0.0, 0.4,  0.0, 1.0 },
  { 0.8, 0.2, 0.6,  0.0, 1.0 },
```

```
  { 0.8, 0.4, 0.1,  1.0, 0.0 },
  { 0.8, 0.6, 0.7,  1.0, 0.0 },
  { 0.8, 0.8, 0.2,  1.0, 0.0 },
  { 0.8, 1.0, 0.8,  1.0, 0.0 },

  { 1.0, 0.0, 1.0,  0.0, 1.0 },
  { 1.0, 0.2, 0.3,  0.0, 1.0 },
  { 1.0, 0.4, 0.8,  1.0, 0.0 },
  { 1.0, 0.6, 0.2,  1.0, 0.0 },
  { 1.0, 0.8, 0.7,  1.0, 0.0 },
  { 1.0, 1.0, 0.0,  1.0, 0.0 }
};

int TestShuffleN[NTEST] = {0}; /* tablou de amestecare */
int TestSXN = {NTEST+1};

/**** NextTestN() - amesteca */
int NextTestN()
{
  int HitsN;
  int wxN;
  int xN, yN;

  if ( TestSXN >= NTEST ) {
    for( wxN = 0; wxN < NTEST; wxN++ )
      TestShuffleN[wxN] = wxN;
    for( HitsN = 0; HitsN < NTEST+NTEST/2; HitsN++ ) {
      xN = rand() % NTEST;
      yN = rand() % NTEST;
      wxN = TestShuffleN[xN];
    TestShuffleN[xN] = TestShuffleN[yN];
    TestShuffleN[yN] = wxN;
    }
    TestSXN = 0;
  }
  return( TestShuffleN[TestSXN++] );
}

/***** TrainNet() - antreneaza reteaua */
int TrainNet( ErrLvlR, MaxPassN )
double ErrLvlR;   /* nivelul de eroare de atins */
long MaxPassN;    /* numarul maxim de pasi */
{
  WREAL rvR[MAX_PES];
  double lsErrR;
  int CurTestN;
  int HitsN;
  long PassN;
  int ExampleN;

  HitsN = 0;
  CurTestN = 0;
  lsErrR = 0.0;
  PassN = 0;
  for( ; ; ) {
    ExampleN = NextTestN();
    lsErrR += Learn( &testE[ExampleN].InVecR[0], &rvR[0],
                     &testE[ExampleN].DoVecR[0], 0.9, 0.5 );
    CurTestN++;
    if ( CurTestN >= NTEST ) {
      PassN++;
      lsErrR = sqrt( lsErrR ) / (double) NTEST;
      if ( lsErrR < ErrLvlR )
        HitsN++;
      else
        HitsN = 0;
      printf( "Pass %3ld Error = %.3f Hits = %d\n", PassN, lsErrR,
              HitsN );
      if ( PassN > MaxPassN || HitsN > 3 )
      break;
      CurTestN = 0;
      lsErrR = 0.0;
    }
  }
  return( 0 );
}

/***** ExplainNet() - executa explicatia si o listeaza */
int ExplainNet( fnP, DitherR )
char *fnP;      /* numele fisierului de iesire */
double DitherR;   /* perturbatia */
{
  FILE *fP;
  int wxN;
  int xN, yN;
  int axN;
  WREAL *wfP;
  static WREAL ivR[MAX_PES] = {0};
  static WREAL ovR[MAX_PES] = {0};
  static WREAL evR[MAX_PES * MAX_PES] = {0};

  if ( *fnP == '\0' )
    fP = stdout;
  else
    if ( ( fP = fopen( fnP , "a" ) ) == (FILE *) 0 ) {
      printf( "Nu pot deschide fisierul de iesire <%s>\n", fnP );
      return( -1 );
    }
  fprintf( fP, "\f\n
          ***Iesirea retelei ca functie de intrarile 1 & 2 ***\n\n" );
  ivR[2] = 0.5;
  for( yN = 20; yN >= 0; yN-- ) {
    if ( ( yN % 2 ) == 0 )
      fprintf( fP, "%6.2f | ", yN / 20. );
    else
      fprintf( fP, "       | " );
    for( xN = 0; xN <= 20; xN++ ) {
      ivR[0] = xN / 20.;
      ivR[1] = yN / 20.;
      Recall( &ivR[0], &ovR[0] );
    if ( fabs( ovR[0] < .1 ) )
        fprintf( fP, "  - " );
    else
        fprintf( fP, "%5.1f", ovR[0] );
  }
  fprintf( fP, "\n" );
  }
  fprintf( fP, "      +-" );
  for( xN = 0; xN <= 20; xN++ )
    fprintf( fP, "-----" );
  fprintf( fP, "\n          " );
  for( xN = 0; xN <= 20; xN++ )
    fprintf( fP, ( xN % 2 ) == 0 ? "%5.1f" : "     ", xN/20. );
  fprintf( fP, "\n" );
  fprintf( fP, "\f\n
          *** Functia de explicatie pentru intrarea 1
          peste domeniul de intrare ***\n\n" );
  ivR[2] = 0.5;
  for( yN = 20; yN >= 0; yN-- ) {
  if ( ( yN %2 ) == 0 )
      fprintf( fP, "%6.2f | ", yN / 20. );
    else
      fprintf( fP, "       | " );
    for( xN = 0; xN <= 20; xN++ ) {
      ivR[0] = xN / 20.;
      ivR[1] = yN / 20.;
      Explain( &ivR[0], &ovR[0], &evR[0], DitherR );
      if ( fabs( evR[0] ) < .1 )
        fprintf( fP, "  - " );
      else
        fprintf( fP, "%5.1f", evR[0] );
    }
    fprintf( fP, "\n" );
  }
```

```
    fprintf( fP, "     +-" );
    for( xN = 0; xN < = 20; xN+ + )
      fprintf( fP, "-----" );
    fprintf( fP, "\n          " );
    for( xN = 0; xN < = 20; xN+ + )
      fprintf( fP, ( xN % 2 ) = = 0 ? "%5.1f" : "     ", xN/20. );
    fprintf( fP, "\n" );
    fprintf( fP, "\f\n
            *** Functia de explicatie pentru intrarea
            2 peste domeniul de intrare ***\n\n" );
    ivR[2] = 0.5;
    for( yN = 20; yN > = 0; yN-- ) {
    if ( ( yN %2 ) = = 0 )
        fprintf( fP, "%6.2f | ", yN / 20. );
      else
        fprintf( fP, "      | " );
      for( xN = 0; xN < = 20; xN+ + ) {
        ivR[0] = xN / 20.;
        ivR[1] = yN / 20.;
        Explain( &ivR[0], &ovR[0], &evR[0], DitherR );
        if ( fabs( evR[2] ) < .1 )
          fprintf( fP, "  - " );
        else
          fprintf( fP, "%5.1f", evR[2] );
      }
      fprintf( fP, "\n" );
    }
    fprintf( fP, "     +-" );
    for( xN = 0; xN < = 20; xN+ + )
      fprintf( fP, "-----" );
    fprintf( fP, "\n          " );
    for( xN = 0; xN < = 20; xN+ + )
      fprintf( fP, ( xN % 2 ) = = 0 ? "%5.1f" : "     ", xN/20. );
    fprintf( fP, "\n" );
    fprintf( fP, "\f\n
            *** Functia de explicatie pentru intrarea 3
            peste domeniul de intrare ***\n\n" );
    ivR[2] = 0.5;
    for( yN = 20; yN > = 0; yN-- ) {
    if ( ( yN %2 ) = = 0 )
        fprintf( fP, "%6.2f | ", yN / 20. );
      else
        fprintf( fP, "      | " );
      for( xN = 0; xN < = 20; xN+ + ) {
        ivR[0] = xN / 20.;
        ivR[1] = yN / 20.;
        Explain( &ivR[0], &ovR[0], &evR[0], DitherR );
        if ( fabs( evR[4] ) < .1 )
          fprintf( fP, "  - " );
        else
          fprintf( fP, "%5.1f", evR[4] );
      }
      fprintf( fP, "\n" );
    }
    fprintf( fP, "     +-" );
    for( xN = 0; xN < = 20; xN+ + )
      fprintf( fP, "-----" );
    fprintf( fP, "\n          " );
    for( xN = 0; xN < = 20; xN+ + )
      fprintf( fP, ( xN % 2 ) = = 0 ? "%5.1f" : "     ", xN/20. );
    fprintf( fP, "\n" );
    if ( fP != stdout )
      fclose( fP );
    return( 0 );
}

/* functia principala */
int main()
{
    int ActionN;
    char *sP;
    char *aP;
    char BufC[80];

    printf( "\nProgram de explicare a concluziilor
            unei retele neuronale\n\n\n" );
    for( ; ; ) {
      printf( "\
C               - creare retea\n\
L [fname]       - incarcare retea antrenata\n\
S [fname]       - salvare retea\n\
P [fname]       - listare retea\n\
F               - stergere retea din memorie\n\
T               - antrenare retea\n\
E [fname]       - explicare retea\n\
X               - iesire din program\n\n\n\
Ce doriti sa faceti ? " );
      fflush( stdout );
      sP = fgets( BufC, sizeof( BufC ) - 1, stdin );
      if ( sP = = (char *) 0 )
        break;
    while( *sP != 0 && *sP < = ' ' )
      sP+ +;
      ActionN = *sP;
      if ( 'A' < = ActionN && ActionN < = 'Z' )
        ActionN -= 'A' - 'a';
      sP+ +;
      for( aP = sP; *aP < = ' '; )
        aP+ +;
      *aP = '\0';
      switch( ActionN ) {
        case 'c':
          BuildNet( 3, 5, 0, 2, 1 );
          break;
        case 'l':
          if ( *sP = = '\0' )
            sP = "network.net";
          LoadNet( sP );
          break;
        case 's':
          if ( *sP = = '\0' )
            sP = "network.net";
          SaveNet( sP );
          break;
        case 'p':
          PrintNet( sP );
          break;
        case 'f':
          FreeNet();
          break;
        case 't':
      TrainNet( 0.003, 100000L );
          break;
        case 'e':
          ExplainNet( sP, .01 );
          break;
        case 'x':
          goto Done;
        default:
          break;
      }
    }
Done:
    return( 0 );
}
```

Administrarea reţelelor

Indiferent ce sistem de operare foloseşte reţeaua, un administrator de LAN are mereu aceleiaşi lucruri de bază de făcut: să creeze noi numere de identificare pentru utilizatori, să întreţină structura de directori şi securitatea file server-ului, să şteargă fişierele vechi, să instaleze şi să actualizeze aplicaţiile, să întreţină cozile de listare, să facă salvări, şi multe atele.

"Sculele" disponibile pentru aceste operaţiuni determină situaţia muncii sale: dacă este vorba de cîteva minute "de iad" de schimbări sau dacă este un efort herculean. Totuşi, atunci cînd o firmă se decide asupra sistemului de operare ce va fi utilizat pentru reţea, adesea sînt neglijate timpul şi efortul cerute de administrarea reţelei.

În acest articol vom examina fiecare din aceste sarcini de administrare şi vom compara modul în care diferă, în ceea ce priveşte susţinerea funcţiei de administrator, cele mai importante sisteme de operare pentru reţele locale - NetWare al lui Novell Inc. LAN Manager al lui Microsoft şi LAN Server al lui IBM.

Crearea de utilizatori şi securitatea

Administratorul unui LAN trebuie să creeze noi numere de identificare pentru utilizatori ori de cîte ori cineva se leagă la reţea şi să şteargă utilizatorii atunci cînd cineva părăseşte reţeaua. Identificatorii pentru utilizatori trebuie să fie adăugaţi în fişierele "batch" la prima instalare a sistemului şi la adăugarea de noi departamente la reţea. Admnistratorii reţelelor de instruire sînt nevoiţi să creeze şi să şteargă identificatori de utilizatori pentru fiecare clasă, în fiecare trimestru. După installarea iniţială, administratorul mai trebuie să adauge utilizatori unul cîte unul atunci cînd sînt angajaţi oameni noi sau cînd sînt transferaţi.

Crearea unui identificator pentru un utilizator nu înseamnă numai asigurarea unui şir de caractere ca nume pentru fiecare persoană care se conectează la reţea. Trebuie să se stabilească şi căror grupuri le va aparţine utilizatorul. Grupurile nu se folosesc numai pentru asigurări la departamente şi grupuri de lucru, ci şi pentru utilizarea aplicaţiilor şi a disponibilităţilor de listare. Dincolo de abilitatea de a gestiona utilizatorii ca grupuri se situează o altă posibilitate valoroasă: gruparea file server-elor şi a altor resurse (de exemplu server-ele de listare) în "domenii". Sistemele de operare pentru reţele care funcţionează în acest mod permit administratorului să stabilească identificatorul utilizatorului o singură dată - pentru domeniu, în loc să fie nevoit să o facă pentru fiecare server în parte. Unele sisteme de acordare de nume extind domeniul "global" (pentru toată întreprinderea).

Un sistem de operare pentru reţea trebuie să pună la dispoziţie un mecanism pentru a crea şi manipula identificatorii de utilizator unul cîte unul cu ajutorul unui utilitar bazat pe meniuri, precum şi un mecanism de gestionare a acestor identificatori pe pachete, pe măsură ce se adaugă noi grupuri de utilizatori sau este nevoie să se facă modificări ce afectează mai mulţi utilizatori. Adăugarea unei noi aplicaţii la 100 de utilizatori - unul cîte unul - poate deveni foarte plictisitoare.

Una din cele mai delicate sarcini ale administratorului unei reţele este stabilirea şi păstrarea controlului asupra accesului utilizatorilor la resursele reţelei. Sub oricare din principalele sisteme de operare pentru reţele, administratorul de sistem poate asigura utilizatorilor privilegii diferite asupra directoarelor, şi uneori chiar asupra anumitor fişiere, de pe file server.

Deoarece toate sistemele au prima linie de apărare bazată pe parole de intrare, este foarte important modull în care sînt protejate fişierele care conţin parolele. Sistemul trebuie să permită un administrator de sistem care să poată citi toate datele de pe server (dacă nu pentru alt motiv, măcar pentru salvări), precum şi un sub-administrator care poate crea utilizatori dar nu le poate citi toate datele. Parolele trebuie să fie memorate în formă criptată, cu verificarea lungimii minime şi a vîrstei maxime ale parolei. De asemenea, sistemul trebuie să dispună de o metodă de limitare a accesului unui utilizator ca staţie sau ca timp.

Cu cît accesul la parola de intrare este mai strict protejat, cu atît mai eficientă va trebui să fie "scula" cu care poate gestiona administratorul acest sistem de securitate. Nici unul dintre sisteme nu asigură acelaşi nivel de protecţie ca securitatea de pe mainframe, dar unele pot asigura o bună securitate a parolei.

Gestionarea aplicaţiilor

Sistemul de operare pentru reţele ideale ar trebui să aibă şi un mecanism simplu de instalare a aplicaţiilor noi. El ar trebui să creeze directoare ale utilizatorilor pentru informaţiile de configurare, să întreţină un sistem de meniuri administrat central - dar nu sufocant şi un sistem de bariere care limitează numărul utilizatorilor care rulează Wondercale la numărul maxim specificat în licenţă.

Un sistem de meniuri administrat central permite administratorului să adauge o aplicaţie în meniurile a 20 de utilizatori fără a-i deranja, şi fără a fi nevoie să-şi părăsească biroul. Atunci cînd utilizatorul "intră" în reţea, trebuie stabilit mediul în care intră. Trebuie să se

asigneze cîte o literă de "drive" fiecărui director de pe file server. Unul sau mai multe din aceste directoare trebuie să fie în "căile" utilizatorului, astfel încît acesta să poată găsi Wonderdraw de pe discul P: fără a trebui să schimbe directorul. Ar putea fi necesar să se lanseze automat, la conectarea în reţea, un sistem de meniuri să se încarce cîteva programe TSR (terminate-and-stay-resident) pentru reţea, cum ar fi un program de poştă electronică sau un controlor de listare.

Odată ce a fost definit acest "mediu" pentru un utilizator, ar trebui să existe o cale de a-l copia la cîţiva utilizatori deodată. Modificările în fişierele cu procedurile de conectare sau în fişierul de comenzi nu ar trebui să îl oblige pe administrator să se "plimbe" pe la staţia de lucru a fiecărui utilizator, nici nu ar trebui să impună utilizarea unui sistem de meniuri specific.

Salvări

Cea mai importantă sarcină a administratorului de reţea este salvarea datelor. Sistemul de operare pentru reţea poate să includă un utilitar de salvare pentru controlul unui dispozitiv de salvare, iar în caz contrar ar trebui să se ferească atît cît este posibil de soft de la alte firme. De asemenea, trebuie să existe posibilitatea ca un utilizator care nu este administratorul sistemului să poată salva toate datele de pe sistem.

Listări

Imprimantele pot să-şi facă de cap pe cont propriu. Dacă o listare care are nevoie de o imprimantă LaserJet cu cartuş PostScript este trimisă din greşeală către o imprimantă Laserjet III fără cartuş PostScript, nu există nici un motiv ca imprimanta să "listeze" 127 de pagini goale, just? Şi totuşi aşa ceva se poate întîmpla în orice reţea.

După stabilirea iniţială a cozilor şi a imprimantelor, cele mai multe din responsabilităţile cu privire la listări ale administratorului se concentrează asupra gestionării cozilor de listare. Un administrator de sistem trebuie să aibă posibilitatea de a şterge un "job" de listare (ca cel menţionat mai sus) din coada de listare înainte de a se produce stricăciuni prea mari, ca şi posibilitatea de a muta listările urgente în vîrful cozii şi de a redirecta coada de la imprimanta laser defectă de la etajul doi către cea de lla etajul trei care funcţionează.

Un "printer spooler" ideal ar trebui să alerteze automat utilizatorii că este necesar să-şi trimită din nou listările. De asemenea, ar trebui să-l anunţe pe administrator cînd are nevoie de hîrtie.

De asemenea, administratorul de sistem va dori să poată transmite unui manager de listări toată complicaţia cu administrarea listărilor fără a compromite securitatea.

NetWare

Crearea de utilizatori şi securitatea

În sistemul NetWare, instrumnetul principal al administratorului pentru gestionarea utilizatorilor este utilitarul axat pe meniuri SYSCON. El poate fi lansat de la orice staţie de lucru DOS din reţea. Cu SYSCON, administratorul poate să creeze utilizatori şi grupuri de lucru, să administreze drepturile lor asupra directoarelor şi să controleze accesul utilizatorilor la sistem. NetWare îi permite administratorului de sistem să limiteze accesul unui utilizator într-o anumită perioadă a zilei, sau la anumite staţii de lucru. De asemenea, administratorul poate fixa cerinţele pentru parole, inclusiv durata maximă şi lungimea minimă, precum şi numărul maxim de staţii de lucru cu care utilizatorul se poate conecta în acelaşi timp. Toate acestea se pot face utilizator cu utilizator.

Administratorul poate crea mai mulţi utilizatori odată şi poate stabili restricţii cu utilitarele MAKEUSER şi USERDEF descrise mai jos. Din nefericire, odată ce utilizatorii au fost creaţi, nu există o metodă de a schimba pe loturi restricţiile de timp şi de parolă. A face modificări pentru 30 de utilizatori, unul cîte unul, pentru a nu le permite accesul în reţea duminica este o cale bună de a-ţi irosi dimineaţa. Deşi SYSCON permite selectarea mai multor utilizatori folosind tasta F5, dacă se selectează mai mult de un utilizator, SYSCON nu permite modificări semnificative.

Sub NetWare 2.1x, numai utilizatorul special "supervisor" - sau echivalentul său din punct de vedere al securităţii - poate crea noi identificatori de utilizator. Deoarece supervisor-ul are drepturi nelimitate asupra tuturor datelor de pe server, numai un număr limitat de utilizatori pot crea alţi utilizatori fără riscuri în ceea ce priveşte securitatea.

NetWare 3.1 şi NetWare 2.2 adaugă tipul de utilizator "şef de grup de lucru". Şeful unui grup de lucru nu are acces la toate informaţiile de pe server, dar poate să creeze noi utilizatori care au drepturi egale, cel mult, cu ale lui.

Şefii grupurilor de lucru pot să şteargă numai acei utilizatori pe care i-au creat sau care le-au fost asignaţi de către supervisor. Aceasta permite administratorului de sistem să delege sarcina de creere a utilizatorilor şefilor de departamente.

Sub NetWare 2.1x administrarea securităţii se face director cu director. Pentru gestionarea securităţii la nivel de director există două utilitare: FILER şi SYSCON, care folosesc meniuri. Există şi alternative cu linie de comandă pentru cei care cred că meniurile ţin numai de restaurante. Şi NetWare 3.1 şi 2.2 permit restricţii de acces la nivel de director. SYSCON permite gestionarea securităţii pe grupuri de utilizatori, dar grupurile trebuie create ţinînd cont de considerentele de securitate.

NetWare nu oferă posibilitatea de a controla accesull utilizatorului la anumite fişiere sau directoare. Unele produse soft, cum ar fi LT Auditor al lui Blue Lance sau LANtight al lui Network Management Inc., oferă această facilitate prin intermediul unor programe TSR pe staţia de lucru.

Utilitarul MAKEUSER produs de NOVELL oferă comenzi pentru configurarea grupurilor de utilizatori. La instalarea unui nou sistem, utilizarea lui SYSCON pentru configurarea a 30 de utilizatori, cinci grupuri şi a tuturor drepturilor lor de acces este o misiune consumatoare de timp. MAKEUSER permite administratorului de sistem să scrie un fişier de comenzi pentru a crea utilizatori, pentru a le crea directoarele, pentru a copia un fişier existent în fişierul lor de comenzi pentru conectare, pentru a controla apartenenţa la grupuri şi pentru a modifica parametri de control ca lungimea parolei şi dreptul de acces la directoarele de pe server.

Cu MAKEUSER, o simplă căutare şi înlocuire poate transforma un fişier de comenzi MAKEUSER care crează noi utilizatori într-unul care şterge utilizatori - sau date din directoarele lor, dacă este necesar. MAKEUSER este util pentru centre de pregătire şi pentru şcoli care trebuie să creeze şi să şteargă conturile elevilor la începutul şi sfîrşitul perioadelor de învăţare.

Deşi MAKEUSER este un instrument preţios, el are cîteva limitări importante. Cu ajutorul lui se pot adăuga utilizatori grupurilor, dar nu se pot crea grupuri şi nici nu se pot adăuga drepturi grupurilor. Deşi se poate crea o bibliotecă de fişiere de comenzi, nu există facilitatea de a memora configuraţii standard pentru utilizatori, iar scrierea fişierelor de comenzi este mult prea apropiată de programare pentru administratorii de sistem care nu au o formaţie tehnică.

Utilitarul USERDEF de sub NetWare 3.1 extinde posibilităţile lui MAKEUSER. El este un program bazat pe meniuri care generează fişiere de comenzi MAKEUSER şi care rulează MAKEUSER. Cu ajutorul lui USERDEF, administratorul de sistem poate construi modele pentru diferite tipuri de utilizatori. De asemenea, USERDEF copiază baza de date cu configuraţia job-urilor de listare a lui PRINTCON pentru utilizatorii definiţi cu ajutorul lui. Utilitarul Origen produs de Preferred Systems Inc. duce mai departe conceptul MAKEUSER, permiţînd unui administrator să construiască o instalare standard de server completă, cu grupuri şi cu structura de directoare agreată de administrator.

NetWare oferă servicii de contabilizare, permiţînd administratorului să modifice timpul de conectare, spaţiul de memorare pe disc şi alte resurse ale sistemului. Din nefericire, contabilitatea lui NetWare nu permite urmărirea numărului de pagini listate - costul cel mai uşor de urmărit într-o reţea şi nu oferă raportări utile.

Gestionarea aplicaţiilor şi a lansării

NetWare include un sistem de creere de meniuri pentru facilitarea instalării aplicaţiilor. Sistemul este destul de primitiv, fără facilităţi de combinare a meniurilor standard cu cele ale utilizatorului sau pentru administrare centrală.

Mediul de conectare al utilizatorului este controlat de o serie de fişiere de comenzi. Sistemul execută automat un fişier de comenzi general - pentru întregul sistem, apoi unul pentru utilizatorul respectiv. Fişierele de comenzi de conectare se aseamănă cu fişierele de comenzi DOS, şi ele pot asigura utilizatorului unităţi de disc şi imprimante. Fişierele de comenzi pentru conectare se editează cu utilitarul SYSCON. Cînd crează noi utilizatori, administratorul poate să copieze fişierul de comenzi de la un utilizator existent, dar utilitarul nu oferă posibilitatea de a scrie un fişier de comenzi pentru un grup.

Salvări

NetWare oferă cîteva utilitare pentru salvări. Dar termenul "utilitare" poate fi greşit folosit, deoarece ele au o utilizare limitată. NetWare Loadable Module pentru salvări pe bandă suportă un set limitat de dispoziţive şi sînt rar utilizate de administratorii de NetWare.

Listări

Cozile de listare sînt gestionate cu utilitarul PCONSOLE, care permite administratorului să controleze job-urile din coada de listare. El nu permite administratorului să controleze imprimantele decît dacă este utilizat programul server de listare PSERVER. Dacă se rulează NetWare 2.1x fără PSERVER, multe din comenzile de control ale imprimantelor trebuie să fie date de la consola file server-ului.

Utilizatorii pot şi ei să ruleze PCONSOLE pentru a-şi gestiona propriile job-uri de listare.

LAN Manager

Crearea de utilizatori şi securitatea

O reţea LAN Manager poate fi administrată de la o staţie de lucru OS/2 sau de la o staţie de lucru DOS care rulează un software avansat pentru staţii. (Staţiile de lucru DOS pot fi instalate fie ca staţii de bază, fie ca staţii avansate. Opţiunea pentru staţie avansată permite administrarea reţelei, dar consumă mai multă memorie).

LAN Manager oferă un utilitar bazat pe meniuri pentru crearea de utilizatori şi grupuri de utilizatori. Administratorul poate stabili orele de conectare, staţiile de lucru permise, directoarele de lucru şi data expirării contului atunci cînd crează utilizatori.

Pot fi stabilite la nivel de server vîrsta maximă şi minimă a parolei, numărul de parole vechi pe care

utilizatorul nu le poate utiliza şi lungimea minimă pentru parolă.

Programul permite administratorului să multiplice un utilizator, copiind toate privilegiile şi proprietăţile pentru un nou utilizator - ceea ce constituie o caracteristică nepreţuită pentru crearea grupurilor. Pentru toate aceste funcţii sînt disponibile şi utilitare care pot fi apelate prin linii de comandă, astfel încît se pot scrie fişiere de comenzi pentru a crea loturi de utilizatori, dar nu există un utilitar destinat acestui scop.

Gestionarea aplicaţiilor şi a lansării

LAN Manager nu oferă un sistem de meniuri administrat central. Fixarea căii pentru fişierele unui utilizator trebuie prin intermediul fişierelor standard de comenzi DOS, iar modificările la o cale existentă se pot face numai la începutul sau la sfîrşitul ei (dar întreaga cale poate fi înlocuită cu o nouă declaraţie de cale).

Un server LAN Manager poate fi instalat fie pentru securitate la nivel de utilizator, fie pentru securitate la nivel de porţiune. Securitatea la nivel de utilizator din LAN Manager funcţionează asemănător cu cea din NetWare. Un utilizator poate avea una sau toate din permisiunile definite (vezi tabela) pentru un director dat.

Securitatea la nivel de porţiune permite asigurarea parolelor şi a permisiunilor pentru fiecare director. Un utilizator care doreşte să asigneze o unitate logică de pe staţia sa de lucru unui director de pe server cu ajutorul comenzii NET USE (similară cu comanda MAP din NetWare) va trebui să introducă parola pentru acel director. Securitatea este asigurată numai prin parola la nivel de director, fără a restrînge accesul la date pentru fiecare utilizator.

LAN Manager cîteva funcţii de control care permit să se supravegheze conectările, deconectările, sau momentele în care utilizatorii asignează o unitate unui director cu comanda NET USE. Nu este supravegheat accesul la nivel de fişier. Un administrator doreşte, de multe ori, să afle cine a lucrat cu un anumit fişier şi cînd. LAN Manager poate urmări cînd a fost asignată o unitate unui director, dar nu indică cînd sau dacă - a fost utilizat un fişier din acel director; de obicei, asignările de directoare se fac în momentul în care utilizatorul se conectează la reţea.

Salvări

LAN Manager se livrează acum cu o copie a softului pentru salvare pe bandă Sytos Plus produs de Sytron - Sytos Plus suportă multe tipuri de unităţi de bandă de pe file server.

LAN Manager-ul lui Microsoft, ca şi varianta LAN Server a lui IBM, mai include şi un interesant serviciu de duplicare. Pentru un file server se poate alege ca, de cîte ori se modifică un fişier dintr-un director specificat, acel fişier să fie copiat în unul sau mai multe directoare de salvare. Directoarele destinaţie pot fi pe un alt file server sau pe o staţie de lucru OS/2. (Pentru reţelele NetWare, o soluţie similară este oferită de produsul LAN shadow furnizat de NMI, care cere o staţie de lucru dedicată).

Serviciile de dupliicare asigură un fel de semi-toleranţă la erori. Ele sînt utile acolo unde pierderea datelor, chiar şi de pe durata unei ore, este inacceptabilă. O limitare o constituie faptul că nu se pot copia fişiere deschise, astfel încît nu se pot proteja baze de date care sînt menţinute deschise pe toată durata zilei de lucru; dar dacă fişierul este închis la un moment dat, serviciul de duplicare îll va copia în directorul de salvare. Seviciull de duplicare poate fi o cale utilă de a distribui actualizări sau alte date de la un server la altul (fişierele noi - sau fişierele existente care au fost modificate - vor fi copiate automat).

Listări

Interfaţa bazată pe meniuri a lui LAN Manager permite administratorului să controleze cozile de listare, să şteargă job-uri şi, în generall, să gestioneze cozile. Utilizatorii şi administratorii pot să oprească temporar, să şteargă sau să elibereze job-uri de listare. LAN Manager îi permite administratorului să protejeze cozile de listare prin parole şi să delege gestionarea cozilor.

De asemenea, LAN Manager oferă administratorului posibilitatea de a proiecta pagini de separare care vor fi listate înaintea fiecrui job pentru identificarea utilizatorului. Paginile de separare pot include comenzi PostScript sau fişiere grafice.

LAN Manager suportă imprimante situate la distanţă şi imprimantepartajate legate la staţiile de lucru ale utilizatorilor astfel încît administratorul nu este nevoit să se deplaseze la file server pentru gestionarea listărilor.

LAN Server

LAN Server este o variantă a lui LAN Manager, dar interfaţa cu utilizatorul şi caracteristicile de administrare sînt diferite.

O reţea LAN Server este administrată de la consola file server-ului sau de la o staţie de lucru OS/2 din reţea. Nu sînt disponibile funcţii de administrare de la staţiile de lucru DOS. După cum este de aşteptat de la un produs IBM, LAN Server oferă un control mai centralizat decît NetWare sau LAN Manager 2.0, punînd la dispoziţie utilitare de gestionare a meniurilor şi un sistem de denumire pe domenii.

Crearea de utilizatori şi securitate

Identificatorii de utilizatori sînt creaţi cu ajutorul programului EE User Profile Maintenance de sub OS/2. Un utilitar se poate conecta numai de la o singură maşină la un moment dat, astfel încît este posibil ca unii utilizatori să aibă nevoie de mai multe conturi. Grupurile

de utilizatori sînt suportate ca în NetWare sau LAN Manager.

Administratorul unei reţele LAN Server asignează fiecărui director o "parcelă". Dacă un utilizator doreşte să asigneze o unitate de disc acelui director, el utilizează comanda NET USE, făcînd referire la "poreclă". Aceasta face utilizarea reţelei mai transparentă, deoarece utilizatorii nu sînt nevoiţi să urmărească unde (pe care server) se află fiecare director.

Totuşi, utilizarea "parcelelor" poate crea necazuri administratorului. De exemplu, dacă un server ajunge la limita capacităţii sale de memorare, ar putea fi necesar să se mute unele directoare pe un alt server. Ar fi bine să existe posibilitatea modificării unei "porecle" existente, pentru a informa LAN Server-ul că acum directorul se află pe un alt server. Din nefericire, nu există posibilitatea modificării numelui server-ului pentru o "poreclă" existentă. Administratorul va trebui să şteargă "porecla" şi să creeze alta cu acelaşi nume.

Programul LAN Requester, de sub OS/2, este unealta de bază pentru administratorul LAN Server. Cu acest utilitar administratorul poate stabili permisiuni ale utilizatorilor şi ale grupurilor pentru "porecle" sau directoare.

Gestionarea aplicaţiilor şi a lansării

Administratorul poate limita numărul utilizatorilor care îşi asignează unităţi la o "poreclă". Dacă există o licenţă pentru patru utilizatori de Wondercalc, se poate limita la patru numărul utilizatorilor pentru directorul Wondercalc. Pentru ca aceasta să aibă efect pentru un produs care are licenţă pentru utilizatori simultani, va trebui ca utilizatorii să-şi asigneze o unitate chiar înainte de a utiliza produsul şi să şteargă asignarea cînd termină.

Programul User Profide Maintenance permite administratorului să stabilească mediul utilizatorului conectat stabilind imprimanta şi asignările de unităţi în momentul conectării şi controlînd ce elemente apar în meniul de selectare a aplicaţiilor al utilizatorului. Din nefericire, adiministratorul nu poate să selecteze cinci utilizatori şi să le adauge tuturor o asignare de unitate.

Modulul DOS LAN Requester din LAN Server este livrat cu un sistem de gestionare a meniurilor, numit selector de aplicaţii, care seamănă cu DOSSHELL din DOS 4.0. Primul meniu acoperă aplicaţiile de pe server - acele aplicaţii care se găsesc pe file server, spre deosebire de cele de pe discul hard al utilizatorului. Acest meniu poate fi controlat de administratorul de sistem. Atunci cînd se adaugă o aplicaţie se pot specifica numai asignările de unităţi care se folosesc în timpul rulării acelei aplicaţii. De asemenea se poate stabili de pe ce unitate se va rula aplicaţia.

Pentru fiecare utilizator se construieşte un fişier PROFILE.BAT care se va executa atunci cînd utilizatorul se conectează la reţea. Calea pentru fişierele unui utilizator poate fi stabilită în PROFILE, dar ea se pierde dacă utilizatorul părăseşte sistemul de meniuri Application Selector. LAN Server nu oferă nici o cale de limitare a folosirii spaţiului pe disc pentru utilizatori sau pentru directoare.

Salvări

LAN Server nu are facilităţi de salvare în afara serviciului de duplicare şi a programelor de salvare şi restaurare din OS/2. Pentru salvări pe bandă, se foloseşte aproape peste tot produsul Sytos Plus.

Listări

Gestionarea cozilor de listare se face prin intermediul OS/2-ului. Un administrator de la o staţie OS/2 poate să şteargă job-uri de listare, să întrerupă temporar imprimante sau cozi şi, în general, să gestioneze procesul de listare. Staţiile de lucru DOS pot să sisteze temporar, să şteargă şi să elibereze job-uri, dar nu pot să oprească cozile de listare.

Care este cel mai bun?

Alegerea unui sistem de operare pentru reţea are un impact teribil asupra vieţii de zi cu zi a administratorului reţelei. Desigur, facilităţile de administrare nu sînt singurele aspecte ale unei reţele care trebuie luate în considerare. Cel puţin la fel de importante sînt viteza, securitatea şi posibilitatea de utilizare a softului provenit din alte surse. Dar acest articol doreşte să se oprească puţin asupra unui aspect care are, de obicei, prioritate minimă: operarea cotidiană a unei reţele.

În ceea ce priveşte facilităţile de administrare nu există un "cîştigător" clar deoarece nu există un beneficiar standard. Dar iată o comparare:

- NetWare are cele mai bune "scule" - de exemplu utilitarul MAKEUSER axat pe fişiere de comenzi. Pe de altă parte, lipsa serviciului de denumire globală şi limitările lui Naming Service din NetWare, transformă administrarea unei reţele vaste într-o corvoadă zilnică.
- LAN Manager are cea mai bine gîndită interfaţă cu utilizatorul. Ea este uşor de folosit şi constituie conduce bine spre epoca GUI. Ca şi la NetWare, serviciul de numiri este slab; mai ales în domeniul accesului independent de amplasare a datelor şi programelor.
- LAN Server are cele mai bune facilităţi de administrare generală, în special din cauza serviciilor sale de numire. Inconvenientul lui LAN Server este că cere o staţie de lucru OS/2 pentru administrarea reţelei.

Howard Marks

(Traducere şi adaptare:Cristian Nagy)

ARTISOFT

O revoluţie în conectarea calculatoarelor

În antiteza faţă de reţelele "clasice" tip client-server (avînd ca cel mai cunoscut şi bine situat pe piaţă firma NOVELL, cu produsul NetWare), unde exista un centru unic în jurul căruia gravitează toate staţiile de lucru, reţelele de tip "egal-la-egal" sînt mult mai democratice, fiecare utilizator putînd deveni, in extremis, un server care să servească pe alţii.

Avantaje

Ce avantaje ar oferi acest sistem radical diferit?

În primul rînd, se elimină necesitatea achiziţionării obligatorii, de la început, a unui calculator server dedicat foarte performant din punct de vedere al vitezei, memoriei RAM şi discului hard, ce este - implicit - şi foarte scump. În al doilea rînd, o staţie de lucru poate să-şi aleagă serverul pe care doreşte (după criterii de performanţă, dotări cu anumite periferice sau aplicaţii soft, spaţiu de manevră pe disc, etc) să-şi execute lucrarea la un moment dat, după cum poate să lanseze în execuţie mai multe lucrări simultan pe servere diferite.

În acelaşi timp, fără a dezafecta nici măcar temporar reţeaua, se poate transforma chiar staţia de lucru respectivă în server sau invers, după cum server poate deveni chiar şi un AT286 "normal", ce poate îndeplini simultan şi funcţii de staţie de lucru. Iată deci că unul din punctele forte ale reţelelor "egal-la-egal" este extrema flexibilitate.

Alte cîteva puncte de rezistenţă sînt memoria RAM necesară extrem de mică, preţul deosebit de scăzut şi extrema simplitate în utilizare.

Aceste tipuri de reţele se pretează foarte bine pentru un număr mic spre mediu de utilizatori. Iniţial reţele locale modeste, pe zi ce trece dezvoltă facilităţi care le apropie de reţelele tip client-server (în privinţa performanţelor tehnice) şi de reţelele de arie largă WAN (în ce priveşte facilităţile de comunicaţie).

Firma

Cel mai dinamic reprezentant al producătorilor de reţele "egal-la-egal" şi care se pare că a reuşit să impună aproape un standard în domeniu este firma americană ARTISOFT, al cărei preşedinte (acţionar principal, căci de un an de zile a devenit companie publică, cotată la bursă), Jack Schoof II, avînd deja aproape 35 de ani (!), a reuşit să crească încasările firmei, bazate în principal pe vînzarea produsului lor LANtastic, numai între sfîrşitul lui 89 şi prima jumătate a lui 91 de la 6,6 milioane $ la peste 41 milioane $, anunţînd închiderea anului fiscal 92 cu peste 73 milioane $ încasări!

Cu o rată de aproape 10% din venituri dedicată publicităţii şi reprezentării în faţa clienţilor, cu fonduri foarte mari puse la dispoziţia cercetării şi proiectării, cu permanente contacte (inclusiv achiziţionări de firme sau cooperări "bombă") în vederea străpungerii de zone în premieră absolută (ultima realizare fiind "puntea" între reţelele de tip DOS şi cele LocalTalk de pe calculatoarele Apple), firma dovedeşte practic că luptă în mod real pentru atingerea scopului propus de preşedintele ei: saltul de la statutul singular de "producător al reţelelor de tip LANtastic" la cel de producător de mijloace de comunicaţie (inclusiv multimedia).

Are însă acest succes financiar şi un suport tehnic real, sau se bazează numai pe o "modă LANtastic"?

Cine eşti dumneata, domnule Schoof ?

După ce la 14 ani a devenit (conform unor mărturisiri făcute de apropiaţi) programator profesionist, s-a apucat de montat calculatoare compatibile IBM PC ("clones") pentru prieteni, afacere ce s-a dovedit rentabilă, banii investindu-i în angajarea unui coleg de şcoală, Alex Karahalios (programator de excepţie, actualmente vicepreşedinte însărcinat cu dezvoltarea şi cercetarea) şi pregătirea începuturilor firmei. ARTISOFT s-a născut în 1982, iar primul lor produs a fost SpiderNet, un produs de partajare a utilizării imprimantelor de către mai multe calculatoare. Produsul era revoluţionar, născut puţin prea devreme, şi cei doi au muncit destul ca să-l impună pe piaţă. Dar, recunosc şi ei, nu a fost un succes comercial deosebit. Dar, avînd această experienţă, începe munca deosebit de tenace de creare a reţelei LANtastic. În 1983, Schoof ia (după propria sa mărturisire) cea mai riscantă decizie din carieră, afectînd 10% din capitalul firmei pentru a plăti o pagină de publicitate în revista BYTE.

Din acest moment începe aventura LANtastic. În 1989, versiunea

3.0 aduce deja aproape 5 milioane de dolari încasări şi se iau decizii mari: dezvoltarea versiunii 4.0, precum şi producerea de componente hard proprii (adaptoare Ethernet şi vocale) cu fiabilitate deosebită (garanţia dată de ei este 5 ani!).

Rezultatul se poate vedea azi pe pieţele întregii lumi, din Japonia şi pînă în SUA, trecînd prin bătrînul continent.

LANtastic

Ce ştie să facă LANtastic 4.0 (care este cel mai răspîndit, cel mai nou şi cu 50% mai rapid fiind însă 4.1)?

Utilizînd circa 12 kB RAM pentru staţiile de lucru şi 40 kB pentru server, este cel mai "mic" nucleu de reţea produs. El se încarcă direct peste sistemul de operare DOS, nu necesită formatarea dedicată a hard-discului şi se poate dezactiva, fără resetarea calculatorului.

Avînd un sistem propriu de cache realizat soft, reuşeşte să obţină performanţe de viteză foarte bune cu transmisie printr-o reţea Ethernet cu fir subţire de 10 MB/s, cu pînă la 300 utilizatori (primit de la început; nu există upgrade-uri!). Atît de bune încît a fost posibilă crearea unei versiuni sensibil mai ieftine, cu plăci adaptoare dedicate şi transmisie prin fir toradat, viteza fiind de numai 2MB/s şi pînă la 32 de utilizatori.

Sistemul de operare se încarcă de pe o singură dischetă (atît în cazul serverului, cît şi a staţiei de lucru), instalare şi "reglajul fin" al reţelei fiind simplificate la maximum şi asistate de meniuri pop-up şi HELP-uri (deocamdată, în limba engleză).

Există şi posibilitatea creării de staţii de lucru fără unităţi de discuri, încărcarea softului de reţea făcîndu-se în mod automat la pornirea calculatorului cuplat în reţea (prin adăugarea unui PROM special pe placa de adaptare). Se pot astfel utiliza, fără rabaturi sesizabile pentru utilizatori, staţii de lucru foarte ieftine la performanţe similare cu un PC complet.

Softul de reţea permite cele mai importante operaţii specifice unor reţele, de la transferul de fişiere şi pînă la lucrul în mod multiutilizator, poşta electronică şi chiar mesagerie vocală (direct sau prin înregistrări digitizate şi stocate pe disc), realizate cu un adaptor special şi un receptor telefonic livrat alături.

De asemenea, un lucru ce nu poate lipsi într-o reţea este securitatea datelor şi a accesului la fişiere şi aplicaţii. În acest doemniu, pe de o parte s-a dezvoltat softul necesar comunicării (prin portul serial) cu un UPS, în privinţa accesului putîndu-se face toate protecţiile cunoscute prin parolă, prin ora (perioada) de lucru şi zona la care se permite accesul.

Există de asemenea aplicaţia care face o supraveghere globală a activităţii în întreaga reţea, putîndu-se ajunge chiar la contorizarea timpului de calculator consumat de fiecare utilizator, în vederea taxării.

Aşa cum aminteam la început, utilizatorii pot alege (dacă li s-a permis accesul în zona respectivă) rularea aplicaţiilor lor pe un server care le asigură cei mai buni parametri urmăriţi (în general viteza de procesare).

Programul spooler de partajare a lucrărilor la imprimante asigură accesul şi execuţia lucrărilor la mai multe imprimante simultan.

Un punct de performanţă deosebită este lucrul cu unităţi de tip CD-ROM, se pare cel mai "curat" utilizat pînă acum în reţele. Dacă unitatea este ataşată unui server, ea poate fi accesată din orice punct al reţelei prin apelare ca un driver obişnuit.

Există compatibilitate pentru rulare de sub Windows ca aplicaţie DOS, dar se produce şi o versiune ce apare ca aplicaţie Windows şi are toate avantajele ce derivă din aceasta. Lucrul sub Windwos se face atît în modul standard, cît şi în cel real şi în cel extins ale procesoarelor 286 şi respectiv 386.

Activitatea de întreţinere a reţelei este extrem de simplă şi nu necesită nici pe departe angajarea expresă a unui inginer de sistem.

Manualul este bine gîndit şi asigură găsirea rapidă a informaţiei dorite, fără a încărca memoria cu detalii inutile. Distribuitorii pot asigura şcolarizări, dar de cele mai multe ori LANtastic se învaţă prin studiu individual şi mai ales prin practica la calculator.

Transformarea într-un server dedicat se face printr-o simplă comandă DOS ("ALONE"), fără a fi necesară decuplarea din reţea sau resetarea întregii reţele. Un server poate gestiona pînă la 5100 de fişiere deschise simultan.

Ca o curiozitate, pe un server se pot lansa pînă la 4 reţele separate simultan!

Dacă se utilizează adaptoare AE-3 (compatibile NE-2000), se pot construi braţe de reţea cablate diferit (coaxial sau torsadat), fără ca, funcţional, reţeaua să sufere cu ceva.

Compatibilitatea porneşte de DOS 3.1, soft-ul de reţea asigurînd blocarea la nivel de înregistrare în cazul accesului multiplu la fişiere.

Este perfect compatibil NetBIOS, deci orice aplicaţie scrisă pentru reţea şi care respectă acest standard va funcţiona sub LANtastic fără nici un fel de probleme.

Un punct puternic este costul, preţul de vînzare (fără taxe) unanim aplicat în lume fiind de 669$ pentru un starter kit ce conţine softul pentru 300 de utilizatori, două plăci adaptoare Ethernet compatibile NE-1000, un cablu subţire gata mufat, în lungime de 8 metri, doi terminatori pentru capetele reţelei, două mufe T, manuale, dischete cu soft. Totul se poate monta în cîteva zeci

de minute (pentru cei foarte neîndemînatici), iar pornirea reţelei durează vreo oră pentru cei ce ştiu un pic de engleză şi nu au văzut sau "pipăit" niciodată vreo reţea.

Distribuitorii de LANtastic au iniţiat o ofertă mult mai bogată de versiuni, componente şi accesorii de reţea.

În ce priveşte partea de conectare oferta ARTISOFT porneşte de la sistemul propriu de 2 MB/s, continuarea cu sistemul clasic Ethernet cu cablu subţire de 10 MB/s (avînd adaptoare compatibile NE-1000 ISA şi Micro Channel, precum şi compatibile NE-2000), cea mai nouă realizare fiind gama completă de adaptoare şi soft pentru cablarea reţelei pe standardul BASE-T (fir torsadat) în sistem Ethernet, ceea ce duce la simplificarea extremă a cablării reţelei.

O versiune interesantă este LANtastic Z, care face legătura doar între 2 calculatoare (de exemplu un desktop şi un notebook), prin intermediul unui cablu serial sau paralel, oferind toate facilităţile reţelei "normale" LANtastic, la preţul de numai 125$!

De asemenea, se furnizează o versiune LANtastic (numai soft), care se poate instala şi cu adaptoare Ethernet produse de alte firme (există drivere pentru multe din cele mai răspîndire tipuri).

O componentă de viitor este adaptorul vocal (ce cuprinde o placă şi un receptor tip telefon), precum şi softul aferent, ce permit deja, pe lîngă mesageria vocală (inclusiv un fel de telefonie de uz intern prin intermediul reţelei) devenită deja clasică la Artisoft, inserarea de mesaje (fragmente) vocale în editoarele de texte, ceea ce reprezintă elemente de multimedia.

Noutăţi

Dintre ultimele realizări ale ARTISOFT putem enumera:

- *LANtastic for Windows* - o versiune ce simplifică foarte mult configurarea reţelei, profitînd în acelaşi timp din plin de grafica şi modul de lucru standardizat al acestui mediu.
- *LANtastic for Netware* - se instalează peste sistemul de operare NetWare 2.1 (declarat "mort" de către NOVELL), extinzîndu-l cu pînă la 300 de utilizatori, la un preţ incomparabil mai mic decît un upgrade NOVELL asemănător şi cu performante aproape identice.
- *LANtastic for APPLE* - face o punte de transfer între reţelele compatibile NetBIOS şi cele LocalTalk, permiţînd accesul reciproc la fişiere, precum şi accesul partajat la resursele de printare.
- *Communication Center* - este un procesor autonom (conţine atît hard, cît şi soft) pentru gestionarea comunicaţiei pe canale seriale şi modem/fax, precum şi cu printere este modulul care permite cuplarea la distanţă cu alte reţele.
- The Network Eye - este un utilitar ce face supravegherea generală a modului de utilizare a reţelei, precum şi alocarea resurselor.

ARTISOFT oferă gama completă de accesorii de conectică pentru instalarea reţelei (mufe, cabluri, HUB-uri, etc.)

Spaţiul nu ne permite, dar am putea prezenta o mulţime de extrase din presa de specialitate care se referă la subiectul LANtastic. Multe firme, de la simple birouri pînă la centrale atomice, chiar firme ce produc ele însele tehnica de calcul (cum ar fi multinaţionala NEC), au analizat piaţa de reţele şi au constatat (chiar practic, prin achiziţionare) că reţelele de tip LANtastic reprezintă cel mai echilibrat compromis între performanţă (de altfel, în permanentă creştere), preţ (deosebit de rezonabil în cazul nostru), costurile de instalare şi întreţinere (aproape inexistente, pentru produsele ARTISOFT) şi dificultatea utilizării (în cazul analizat în acest material, orice operator ce ştie să dea comenzi în DOS este pregătit să facă acelaşi lucru şi pentru reţea).

În opinia autorului, reţelele LANtastic constituie exact veriga care lipseşte între dorinţa de evoluie către conectivitate a masei de calculatoare achiziţionate în ultimii trei ani în România, şi buzunarul destul de sărac al românului, dar fără a afecta funcţionalităţile dorite.

Un principiu important în alegerea reţelelor spune: "aveţi mare grijă să nu confundaţi dorinţele dvs. relativ la reţele cu necesităţile reale ale aplicaţiilor ce trebuie deservite. Păgubit va fi numai buzunarul celui ce plăteşte". Gîndiţi-vă că buzunarul ce plăteşte e, totuşi, al românului.

ing. OCTAVIAN CUDALBU

Un tabel de control pentru succes

Ce trebuie pretins de la un sistem de gestionare a desenelor/documentelor

Odată cu apariţia reţelelor, beneficiile utilizării unei forme oarecare de sistem de gestionare a desenelor/documentelor (DMS) devin mai clare. Utilizatorii de CAD şi managerii sînt confruntaţi cu o multitudine de factori de influenţă, incluzînd tehnologiile în schimbare, un număr în creştere de utilizatori, un set tot mai mare de desene şi documente conexe, cereri ale clienţilor şi o piaţă competiţională într-un climat economic foarte strîns.

Fiind daţi aceşti factori, cum ar trebui o întreprindere să procedeze în identificarea şi implementarea unei soluţii DMS?

Prima şi a doua generaţie DMS

Aşa cum am constatat în cazul altor aplicaţii (de ex. spreadsheet-uri şi baze de date), sofisticarea şi funcţionalitatea programelor DMS evoluează. În lumea mediilor de dezvoltare a bazelor de date relaţionale se aude tot mai frecvent de a patra generaţie de limbaje (4GL). Tot astfel, în noul domeniu al DMS-urilor a existat o primă generaţie, urmată de o a doua în plină dezvoltare ca rezultat a doi factori critici: trecerea de la staţii de lucru independente către configuraţii de reţea şi capacitatea de lucru în reţea a AutoCAD Release 11, care oferă mecanismele de bază pentru protecţia fişierelor în reţea.

Programele de gestionare a desenelor nu sînt o noutate. Ele sînt deja disponibile de cîţiva ani, fiind caracterizate de funcţii ca localizarea desenelor, examinarea lor indiferent de programul CAD din care provin şi forme simple de adnotare a lor (ca numele proiectului, descrierea, data ultimei utilizări etc.). De fapt unele dintre aceste sisteme timpurii erau denumite "de vizualizare". Ele erau în mod normal suficiente pentru utilizatori individuali operînd pe o staţie de lucru izolată cu un număr uşor manevrabil de desene şi nevoi reduse de partajare a acestora şi a datelor conexe.

În cadrul activităţilor CAD în reţea apar probleme care nu se întîlnesc la nivelul PC. Cu siguranţă, aplicaţiile au fost dezvoltate în vederea rezolvării problemelor evidente de protecţie a fişierelor şi chiar de autorizare a utilizatorului concurent. Oricum, operaţiu-

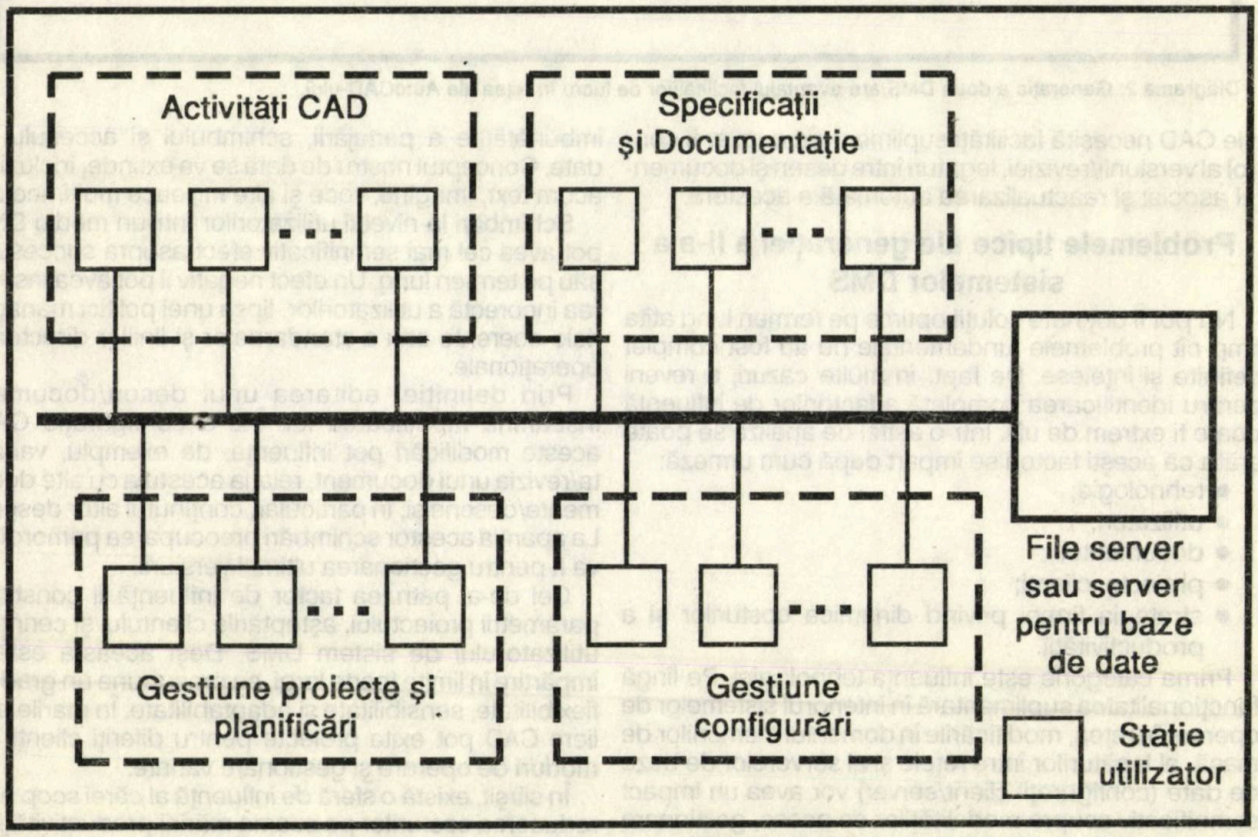


Diagrama 1: *Sistemele DMS au oferit de-a lungul anilor funcţii ca localizarea desenelor, examinarea şi extragerea acestora indiferent de programul CAD care le-a creat precum şi forme simple de adnotare a lor.*

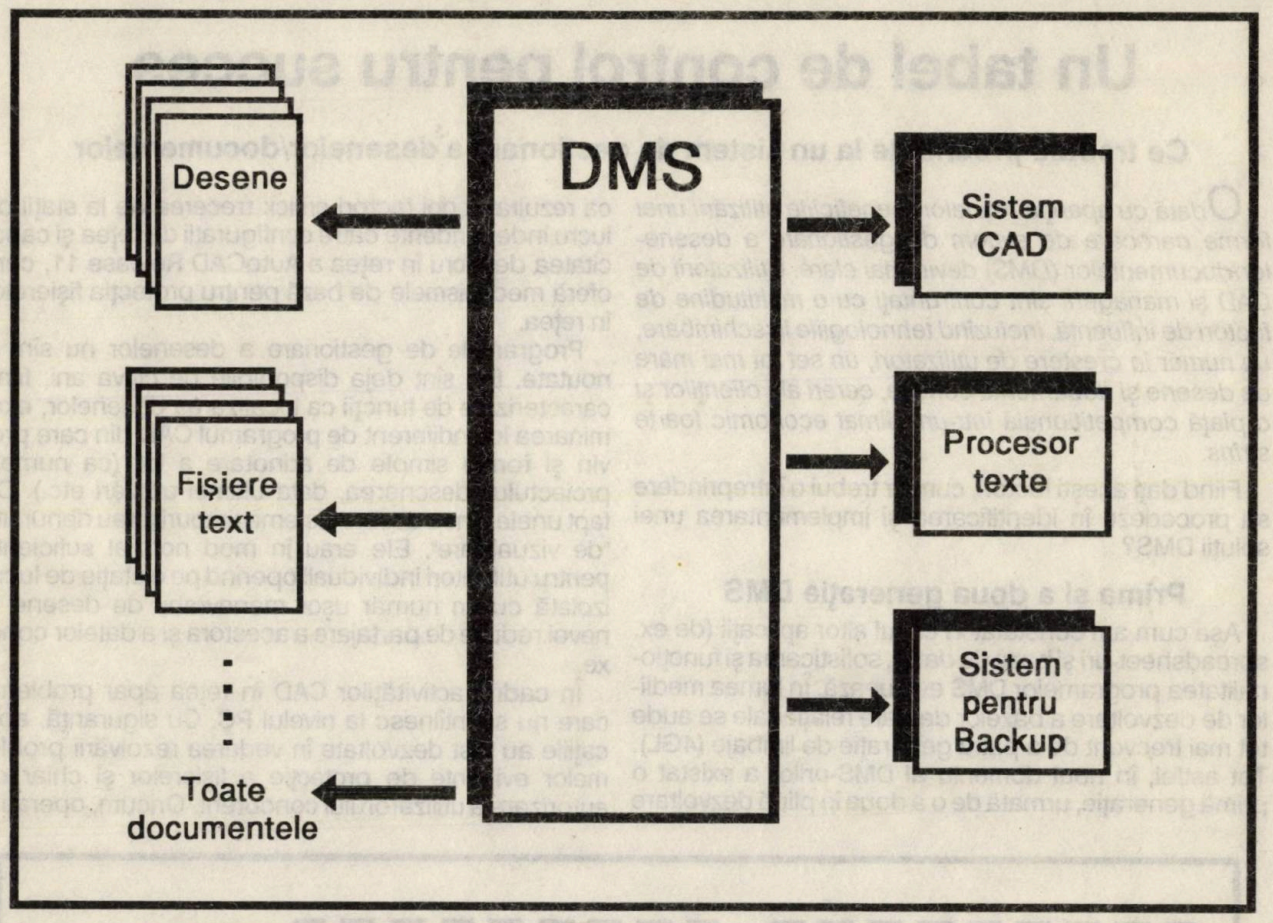


Diagrama 2: Generaţia a doua DMS are avantajul facilităţilor de lucru în reţea ale AutoCAD-ului.

nile CAD necesită facilităţi suplimentare cum ar fi control al versiunii/reviziei, legături între desen şi documentul asociat şi reactualizarea automată a acestora.

Problemele tipice ale generaţiei a II-a a sistemelor DMS

Nu pot fi obţinute soluţii optime pe termen lung atîta timp cît problemele fundamentale nu au fost complet definite şi înţelese. De fapt, în multe cazuri, a reveni pentru identificarea completă a factorilor de influenţă poate fi extrem de util. Într-o astfel de analiză se poate arăta că aceşti factori se împart după cum urmează:

- tehnologie;
- utilizatori;
- documente;
- proiecte, clienţi;
- strategia firmei privind dinamica costurilor şi a productivităţii.

Prima categorie este influenţa tehnologiei. Pe lîngă funcţionalitatea suplimentară în interiorul sistemelor de operare în reţea, modificările în domeniul memoriilor de masă, al legăturilor între reţele şi al serverelor de baze de date (configuraţii client/server) vor avea un impact semnificativ asupra modalităţilor de acces, gestionare şi control al desenelor/documentelor. Aceste tehnologii vor deschide larg porţile spre eforturi de îmbunătăţire a partajării, schimbului şi accesului la date. Conceptul nostru de dată se va extinde, incluzînd acum text, imagine, voce şi alte mijloace multimedia.

Schimbări la nivelul utilizatorilor într-un mediu DMS pot avea cel mai semnificativ efect asupra succesului său pe termen lung. Un efect negativ îl pot avea instruirea incorectă a utilizatorilor, lipsa unei politici manageriale coerente sau a standardelor şi liniilor directoare operaţionale.

Prin definiţie, editarea unui desen/document înseamnă modificarea lui. Într-o configuraţie CAD, aceste modificări pot influenţa, de exemplu, varianta/revizia unui document, relaţia acestuia cu alte documente/desene şi, în particular, conţinutul altor desene. La apariţia acestor schimbări preocuparea primordială va fi pentru gestionarea ultimei versiuni.

Cel de-al patrulea factor de influenţă îl constituie parametrii proiectului, aşteptările clientului şi cerinţele utilizatorului de sistem DMS. Deşi aceasta este o împărţire în limite foarte largi, ea presupune un grad de flexibilitate, sensibilitate şi adaptabilitate. În marile ateliere CAD pot exita proiecte pentru diferiţi clienţi, cu moduri de operare şi gestionare variate.

În sfîrşit, există o sferă de influenţă al cărei scop este reducerea costurilor pe seama măririi productivităţii, a optimizării şi a atingerii unui nivel înalt al calităţii. Accentul în creştere pus pe reţele şi arhitecturi de tip

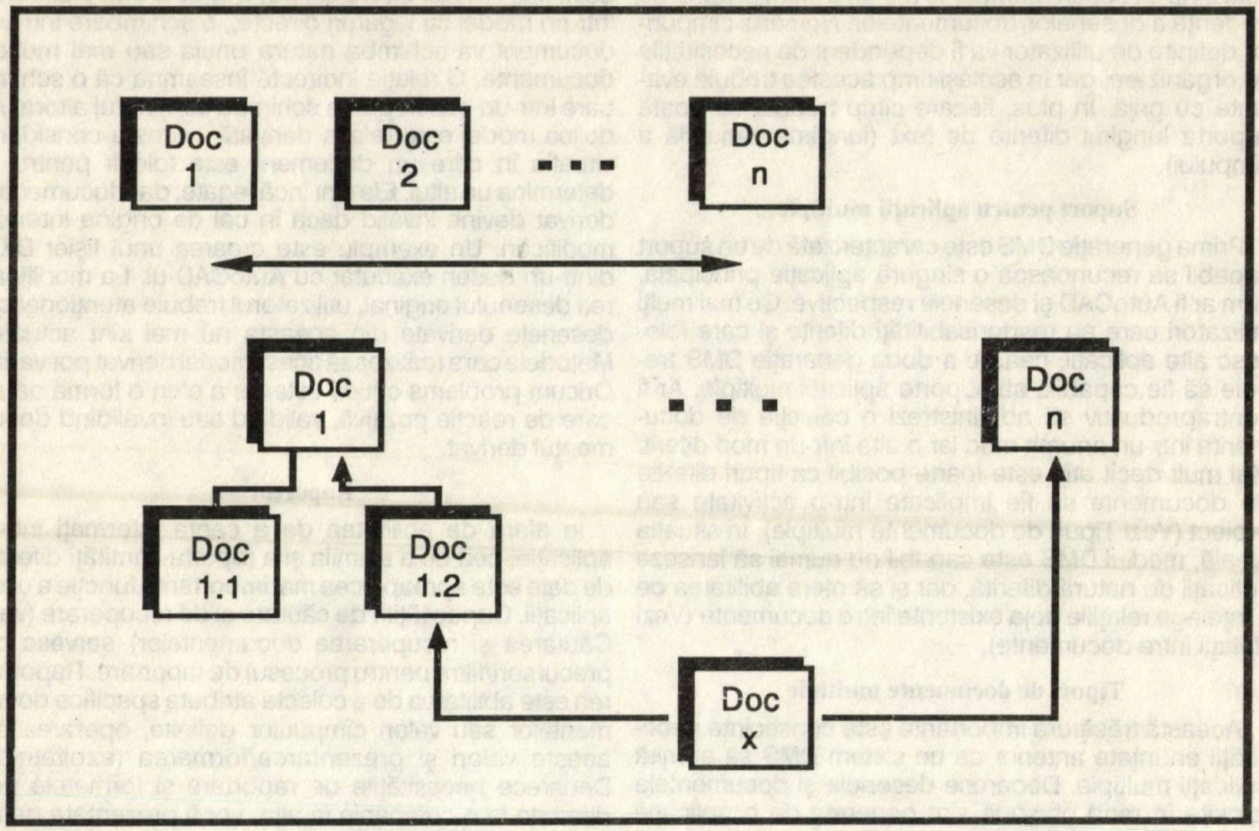


Diagrama 3: Într-un Sistem DMS, pe o relaţie orizontală, modificări într-un document vor afecta un altul. Pe o relaţie verticală, modificările documentului nu vor afecta în mod necesar un altul.

client/server nu lasă nici un dubiu că dimensionarea minimală acaparează rapid atenţia managerilor. Aceste influenţe ale tehnologiei, la care se adaugă mărirea independenţei utilizatorilor, necesită o reevaluare a modalităţilor de atingere a scopurilor unei anumite întreprinderi.

Acestea sînt toate cerinţe foarte importante influenţînd orice mediu DMS. Oricum, se pot articula soluţii răspunzînd nevoilor de astăzi ale unui mediu CAD în reţea şi pregătind adaptarea la viitoare schimbări.

Soluţiile oferite de cea de-a doua generaţie

La succesul opraţiilor unui sistem de gestiune a documentelor sînt necesare următoarele facilităţi:

Suportul oferit de o configuraţie în reţea.

Deşi o aplicaţie poate opera eficient în reţea, există două zone funcţionale importante care trebuie să fie disponibile într-un mediu DMS: posibilitatea de a bloca în prealabil accesul la un fişier şi/sau înregistrare pe durata utilizării şi, în al doilea rînd, de a permite drepturi suplimentare de acces în reţea. Capacitatea de protejare a fişierelor este primordială. Aceasta impune că, o dată ce un document a fost folosit într-o aplicaţie, alţi utilizatori nu pot reface sau salva acel document pînă cînd utilizatorul care l-a refăcut anterior nu l-a eliberat. În acest caz, sistemul de administrare a desenelor trebuie să prevadă o condiţie de stare, pentru a identifica cine, cînd, ce şi unde a accesat documentul. Ideea drepturilor suplimentare de acces în reţea implică posibilitatea ca un mediu DMS să poată fi configurat să creeze (nu să elimine) restricţii ulterioare. De exemplu dacă un utilizator din reţea lucrează la proiectul A, atunci el poate accesa doar documentele asociate acelui proiect, cu toate că drepturile lui de utilizare a reţelei sînt mai largi.

Caracteristicile documentelor, adnotări şi informaţii descriptive

Oricărui document îi sînt asociate informaţii care descriu conţinutul acestuia, modul lui de folosire, legăturile cu proiectul, precum şi alte caracteristici. Aceste informaţii de tip text sînt externe documentului. Caracteristicile descriptive sînt nu numai comode cînd se trece în revistă "profilul" unui document, dar pot avea o valoarea semnificativă în momentul în care se încearcă regăsirea documentelor pe baza anumitor condiţii. Atributele documentului sau cîmpurile sînt folosite pentru a stoca aceste informaţii. Un sistem DMS ideal ar trebui să permită o combinaţie de cîmpuri predefinite şi cîmpuri definite de utilizator. Această ultimă trăsătură este indispensabilă atîta timp cît fiecare companie are un sistem propriu de evidenţă a desenelor/documentelor. Numărul cîmpurilor definite de utilizator. Această ultimă trăsătură este indispensabilă

atîta timp cît fiecare companie are un sistem propriu de evidenţă a desenelor/documentelor. Numărul cîmpurilor definite de utilizator va fi dependent de necesităţile de organizare, dar în acelaşi timp acestea trebuie evaluate cu grijă. În plus, fiecare cîmp trebuie să poată suporta lungimi diferite de text (lungime variabilă a cîmpului).

Suport pentru aplicaţii multiple

Prima generaţie DMS este caracterizată de un suport capabil să recunoască o singură aplicaţie principală, cum ar fi AutoCAD şi desenele respective. Cu mai mulţi utilizatori care au responsabilităţi diferite şi care folosesc alte aplicaţii, cea de-a doua generaţie DMS trebuie să fie capabilă să suporte aplicaţii multiple. Ar fi contraproductiv să administrezi o colecţie de documente într-un anumit mod iar o alta într-un mod diferit. Mai mult decît atît, este foarte posibil ca tipuri diferite de documente să fie implicate într-o activitate sau proiect (Vezi Tipuri de documente multiple). În situaţia ideală, mediul DMS este capabil nu numai să lanseze aplicaţii de natură diferită, dar şi să ofere abilitatea de a înţelege relaţiile deja existente între documente (Vezi Relaţii între documente).

Tipuri de documente multiple

Această trăsătură importantă este consecinţa necesităţii enunţate anterior ca un sistem DMS să admită aplicaţii multiple. Deoarece desenele şi documentele folosite în mod obişnuit sînt generate de o aplicaţie specifică, sistemul DMS trebuie să înţeleagă această legătură pentru a putea invoca aplicaţia corectă.

În mod similar, necesitatea de a suporta desene şi documente produse în exteriorul calculatorului este critică dacă o organizaţie menţine un număr mare de documente şi desene pe hîrtie. Conversia acestora într-un format electronic este de cele mai multe ori nepractică şi chiar imposibilă. Cu un sistem DMS capabil să suporte acest tip de document, toate celelalte funcţii DMS, cum ar fi conexiunile din document, (vezi Relaţiile dintre documente) pot funcţiona.

Relaţiile dintre documente

Cu un mediu DMS care asigură suportul pentru tipuri multiple de aplicaţii şi documente, abilitatea de a specifica, menţine şi înţelege relaţiile dintre documente devine un instrument puternic şi important pentru aplicaţiile generale de administrare a documentelor. Fişierul de referinţă externă din AutoCAD Release 11 permite utilizatorilor accesul la desenele existente fără o scriere fizică a entităţilor acestuia în structura unui alt desen. Avantajul, desigur, este că aceste fişiere de referinţă sau desene folosite în alte desene sînt definite o singură dată. Preocuparea evidentă priveşte posibilităţile DMS de a face cunoscut utilizatorului ce desene vor fi afectate dacă un fişier de referinţă va fi modificat sau şters.

Exemplul precedent care foloseşte fişiere de referinţă evidenţiază unul din cele două modele de relaţii dintre documente ce pot fi aplicate.

Exemplul este privit ca un model "de conexiuni". În acest caz, un document este legat cu altul. Mai mult decît atît, relaţiile dintre ele pot fi directe sau indirecte. Într-un model cu legături directe,, o schimbare într-un document va schimba natura unuia sau mai multor documente. O relaţie indirectă înseamnă că o schimbare într-un desen poate schimba conţinutul altora. Al doilea model este relaţia derivată - acesta consideră situaţia în care un document este folosit pentru a determina un altul. Ele sînt încă legate, dar documentul derivat devine invalid dacă în cel de origine intervin modificări. Un exemplu este crearea unui fişier DXF dintr-un desen executat cu AutoCAD-ul. La modificarea desenului original, utilizatorul trebuie atenţionat că desenele derivate din aceasta nu mai sînt actuale. Metodele care realizează acest model derivat pot varia. Oricum problema critică este de a oferi o formă oarecare de reacţie pozitivă, validînd sau invalidînd documentul derivat.

Raportul

În afară de abilitatea de a capta informaţi într-o aplicaţie, cea de a asimila şi a raporta cantităţi diferite de date este aproape cea mai importantă funcţie a unei aplicaţii. Capacităţile de căutare şi de recuperare (vezi Căutarea şi recuperarea documentelor) servesc ca precursori/filtre pentru procesul de raportare. Raportarea este abilitatea de a colecta atribute specifice documentelor sau valori cîmpurilor definite, operarea cu aceste valori şi prezentarea/formarea rezultatelor. Deoarece necesităţile de raportare şi formatele pot diferi de la o companie la alta, vor fi prezentate patru categorii diferite de metode de raportare. O combinaţie a acestora poate fi utilizată pentru a asigura majoritatea necesităţilor de raportare. Acestea sînt:

- Ad-hoc: abilitatea de interogare în limba engleză.
- SQL (Structured Query Language): limbaj structurat de interogare - similar abordării Ad-hoc dar care foloseşte un standard industrial. Puterea sa constă în posibilitatea abordării Ad-hoc dar care foloseşte un standard industrial. Puterea sa constă în posibilitatea de a interoga mai multe fişiere de date la fel de bine ca şi servere pentru baze de date.
- QBE (Query By Example): se concentrează asupra prezentării vizuale a coloanelor (cîmpurilor) disponibile, permite selecţia cîmpurilor pentru a fi afişate şi specifică criteriile condiţionale dorite. Avantajul acestei abordări este că "se poate vedea ceea ce se va obţine" ("see-what-you-will-get").
- Programe de scriere de rapoarte: tind să combine o abordare QBE cu un domeniu larg de opţiuni de formatare. Unele vor include funcţionalitate SQL în care funcţiile ad-hoc sînt transparente utilizatorului final.

Facilităţi de căutare şi recuperare a documentelor

Măsura critică a oricărui sistem DMS este abilitatea de a găsi şi recupera documente indiferent de locaţie, nume şi orice alte caracteristici ale lor(vezi Caracteristicile documentelor). Abilitatea de căutare trebuie să admită un număr nelimitat de clauze condiţionale şi operatori folosiţi la crearea criteriilor de căutare. Clauzele condiţionale sînt condiţiile "AND" şi "OR" ("ŞI"/"SAU" condiţional) al unei operaţii. Operatorii includ funcţiile Booleene cum ar fi egalitatea, mai mic,

mai mare, incluziunea, apartenenţa la un domeniu etc. Criteriul ideal de recuperare îl reprezintă abilitatea de căutare pe o condiţie de "configuraţie fonetică asemănătoare". Trebuie găsit de exemplu un document cu numele apropiat de "TWO". Rezultatul poate fi reprezentat prin şirurile de caractere următoare "2", "TO", "TOO" şi desigur "TWO".

Controlul evoluţiei etapelor de prelucrare a unui document

Urmărirea şi prezentarea istoricului activităţilor legate de un document va fi unul din instrumentele manageriale cele mai puternice pentru reducerea costurillor şi creşterea productvităţii şi calităţii. Istoricul unui document ar trebui să includă data, durata, utilizatorul, staţia de lucru şi activităţile respectiv operaţiile executate asupra documentului. Activităţile ce vor fi înregistrate trebuie să conţină accesul, modificările, copierea, mutarea, redenumirea, printarea, plotarea, etc. În plus, poate fi util ca schiţele şi notele scrise cu mîna să fie asociate acestor activităţi la dorinţa utilizatorului. Cu posibilităţi de configurare după dorinţă, pot fi integrate şi alte aplicaţii conexe, ca estimarea timpului de lucru, evidenţa materialelor şi a costurilor, analiza în timp a fluxului tehnologic sau suportul de gestiune a configurărilor.

Medii configurabile de utilizator

Cu cît mai mulţi utilizatori sînt legaţi la o reţea cu atît va creşte gama de discipline, funcţii, consideraţii de acces şi responsabilităţi. Ca o consecinţă, utilizatorii pot întîlni desene, proiecte, coduri şi convenţii precum şi alte colecţii operaţionale de date care sînt irelevante şi/sau care nu sînt cerute de propriile sarcini. Oferind posibilitatea de configurare a mediului, operatorii au acces la informaţii specifice sarcinilor şi ariei lor de responsabilităţi. Aceste "sesiuni" pot defini aplicaţia implicită curentă, locul în care sînt salvate desenele, tipul documentelor încărcate la pornire, condiţiile de acces şi restricţiile.

TABEL DE CONTROL PENTRU EVALUAREA DMS

Caracteristici/Funcţii	Pondere	Produs A		...	Produs n		
		Notă	Scor		Notă	Scor	
Exemplu	8¤	9¤	72		6¤	48	
Suport pentru reţea							
Caracteristici document							
Suport pentru aplicaţii multiple							
Tipuri de documente multiple							
Relaţii între documemte							
Raportare							
Căutare şi extragere documente							
Control evoluţie prelucrare doc.							
Medii configurabile de utilizator							
Convenţii denumire documente							
Verificare intrări/ieşiri							
Planificare tehnologică							
Arhivare							
Sisteme de gestiune baze de date							
Utilizare surse de date existente							
Adaptare de către utilizator							
Elemente specifice utilizatorului							
Element 1							
Element 2							
Element 3							
SCOR TOTAL							

¤ Folosiţi valorile 0-10 unde 10 reprezintă valoarea maximă acordată ponderii importanţei acordate unei caracteristici DMS respectiv notei de evaluare a produsului.

Tabel 1. Alegerea DMS-ului potrivit.

Evaluînd corect factorii şi cerinţele prevăzute în acest tabel, managerul de reţea poate lua o decizie, bazată pe condiţii exacte, derivate din necesităţile companiei referitor la sistemul de gestiune a documentelor DMS.

Convenţii de denumire a desenelor/documentelor

O limită a DOS-ului stabileşte numărul de caractere ale numelui unui fişier la opt. Pe de altă parte, multe sisteme de organizare folosesc coduri ce depăşesc această limită de opt caractere. În astfel de cazuri un sistem de organizare DMS trebuie să înlăture acest neajuns. Ca o extensie la aceasta, este capacitatea unui mediu DMS de a începe un nou document prezentînd automat utilizatorului nume implicite succesive, într-o secvenţă controlată. Aceasta ar fi soluţia ideală pentru sistemul de codificare all propriei firme,, desene multiple cu număr de pagini secvenţial, etc.

Verificarea intrărilor/ieşirilor

În mediile în care modificările desenelor/documentelor nu pot fi făcute în mod arbitrar, trebuie să existe metode de verificare a intrărilor şi ieşirilor. Verificarea intrărilor/ieşirilor este în primul rînd procesul de plasare a unei copii a unui document într-un spaţiu de lucru în scopul trecerii în revistă şi/sau modificării, de posibillă trecere spre alt utilizator pentru verificare/aprobare şi apoi replasare a documentului în zona de stocare iniţială. Acest proces trebuie să prevină inadvertenţele sau schimbările nedorite şi să controleze accesul simultan la acelaşi document de către mai mulţi utilizatori. Indiscutabil, semnarea şi aprobarea unui document este un proces lipsit de rigoare, standarde şi convenţii. Unele dintre aceste lipsuri apar datorită deficienţelor tehnologice actuale.

Planificarea tehnologică, programarea activităţilor şi conducerea evenimentelor

Fluxul tehnologic şi planificarea sarcinilor de lucru a fost multă vreme o aplicaţie separată, situată adesea la nivelul aplicaţiilor de conducere a proiectelor. Oricum, cu mai mulţi utilizatori operînd în interiorul unei reţele, această funcţiune poate fi un avantaj foarte important într-un mediu DMS. Cînd documentele parcurg o serie de stadii este ideal să ai posibilitatea să repartizezi/planifici o activitate pentru un anumit operator sau document. De fapt cele mai multe organizaţii au deja conturate şi stabilite stadiile şi operaţiile. Aceasta poate include atribuiri de sarcini specifice individuale sau atribuiri globale în scopul analizării. De exemplu, după terminarea unui desen, acesta poate fi înaintat pentru operaţia de verificare/analizare. Ca şi în schimbul de mesaje dintr-o reţea, atribuirea va fi notificată ca o activitate nerezolvată. Această notificare poate apărea ca un mesaj pe ecran sau sub forma unei liste de evidenţă a activităţilor ce trebuie făcute. Această trăsură asigură desenelor integritatea; identifică în ce stadii se află desenele, cine le are şi cît timp au rămas în anumite stadii permite o mai bună înţelegere a operaţiilor CAD şi a altor resurse (plottere, dispozitive pentru efectuare backup-ului, etc).

Arhivarea

Odată ce o colecţie de desene sau documente pentru un anumit proiect este completă o operaţie de arhivare poate elibera spaţiu pe disc prin înlăturarea informaţiei care nu mai este actuală. În unele cazuri, după arhivare poate apărea necesitatea de a restaura aceste documente precum şi datele conexe pentru o actualizare a informaţiei. Asociate unui desen sau document gestionat într-un mediu DMS vor fi adnotarea informaţiei, istoricul, legăturile posibile cu alte documente. La arhivare trebuie mutat nu numai documentul actual, ci şi datele asociate acestuia.

Sisteme de gestiune a bazelor de date DBMS

Funcţiunile de bază ale unui DMS - gestiunea documentelor şi datelor asociate acestora, înzestrarea cu opţiuni extinse de interogare şi raportare şi asigurarea platformei pentru mai mulţi utilizatori - presupun utilizarea unui sistem de gestiune a bazelor de date (Database Management System - DBMS). Platforma DBMS este temelia unei soluţii DMS. Deşi există alte abordări, ele tind să fie soluţii bine determinate şi sînt proiectate în general pentru o aplicaţie strict specifică (sisteme dedicate), cu flexibilitate minimă. Pe măsură ce cerinţele unei organizaţii se schimbă şi tehnologiile hardware şi software se îmbunătăţesc, aceste abordări strict specifice vor fi rapid abandonate, atingîndu-şi limitele. Deşi aşa numitele modele de baze de date "orientate pe obiect" asigură o comprehensibilitate superioară, DBMS-ul relaţional rămîne în mod cert modelul ce trebuie ales.

Utilizarea surselor de date deja existente

Un sistem DMS va fi rar implementat într-un mediu în care nu există date definite anterior care să se constituie ca suport pentru controlul şi gestiunea documentelor. Acestea pot fluctua de la un simplu sistem de numerotare a desenelor pînă la o aplicaţie complexă cum ar fi un mediu de gestiune complet configurat. În orice caz aceste surse de date existente pot constitui un avantaj ajutînd la menţinerea unei investiţii primare în organizarea structurilor de date existente şi în aplicaţiile asociate lor.

Modul ideal de abordare ar fi abilitatea mediului DMS de a citi şi scrie în aceste surse de date externe. Aceasta elimină nevoia importului/exportului de date şi asigură că datele folosite sînt cele mai recente. Unele sisteme oferă posibilitatea de scriere/citire din programele obişnuite de baze de date şi calcul tabelar.

Adaptarea organizării DMS la cerinţele utilizatorului

De cele mai multe ori ceea ce doriţi să faceţi astăzi nu coincide cu ceea ce veţi dori ulterior. Cum am spus mai sus, există prea multe influenţe pentru ca marea majoritate a aplicaţiilor să rămînă nemodificabile. Nevoia de a adapta o anumită organizare va fi determinată de imposibilitatea ca aceasta să rămînă în limitele unui sistem cu arhitectură închisă. Adaptabilitatea înseamnă abilitatea de a modifica meniul, fereastra pentru introducerea datelor şi operaţiilor precum şi raportarea. Managerul reţelei trebuie să se informeze ce funcţii adiţionalei pot fi dorite/cerute mai tîrziu, şi dacă mediul DMS poate fi adaptat la modificări ulterioare.

Direcţii de orientare: abordare practică

Acum după ce direcţiile de orientare au fost stabilite, vom sugera un model de lucru (tabelul 1) pentru cei care intenţionează să înceapă sau sînt implicaţi în identificarea şi alegerea unui sistem DMS pentru propria întreprindere. Cu toate că există multiple elemente ce trebuie luate în considerare, unele pot fi mai puţin importante sau chiar inutile. În plus, pot apare şi alte consideraţii importante ce trebuie reţinute. Folosind tabelul 1, atribuiţi o pondere corespunzătoare importanţei fiecărui element precizat şi adăugaţi alte elemente specifice necesităţilor proprii. Odată stabilite aceste ponderi, evaluaţi versiunile sistemelor DMS candidate, comparînd cu rezultatele din tabel ("checklist"). Puneţi o notă fiecărei componente evaluate şi înmulţiţi-o cu ponderea atribuită. Odată obţinut totalul pentru fiecare sistem, aveţi deja determinat cel mai bun sistem corespunzător necesităţilor, după luarea în consideraţie a costului, specificul activităţii, posibilităţilor de instruire şi a altor factori.

Traducere şi adaptare după articolul "A checkllist for success" publicat de Martin M. van der Roest în revista CADENCE realizate de Vincenţiu Popescu

Rezumat

Spre deosebire de alte aplicaţii ce produc diverse forme de rezultate tangibile, un sistem DMS aduce beneficii sub forma creşterii eficienţei şi productivităţii. Pe de altă parte dacă sistemul este inadecvat sau impropriu implementat rezultatele pot fi foarte costisitoare.

Acest tabel de control este similar cu stabilirea unui plan de vacanţă. Stabiliţi mai întîi ce doriţi să faceţi, itinerariul, etc. Rezultatele unui plan reuşit se vor materializa într-o plăcută şi memorabilă experienţă. În acelaşi mod, cu o atentă evaluare şi implementare, un sistem DMS vă va produce profituri pe termen lung. Astfel veţi avea posibilitatea să vă concentraţi asupra afacerilor şi vă va permite să vă ocupaţi de ceea ce ştiţi cel mai bine.

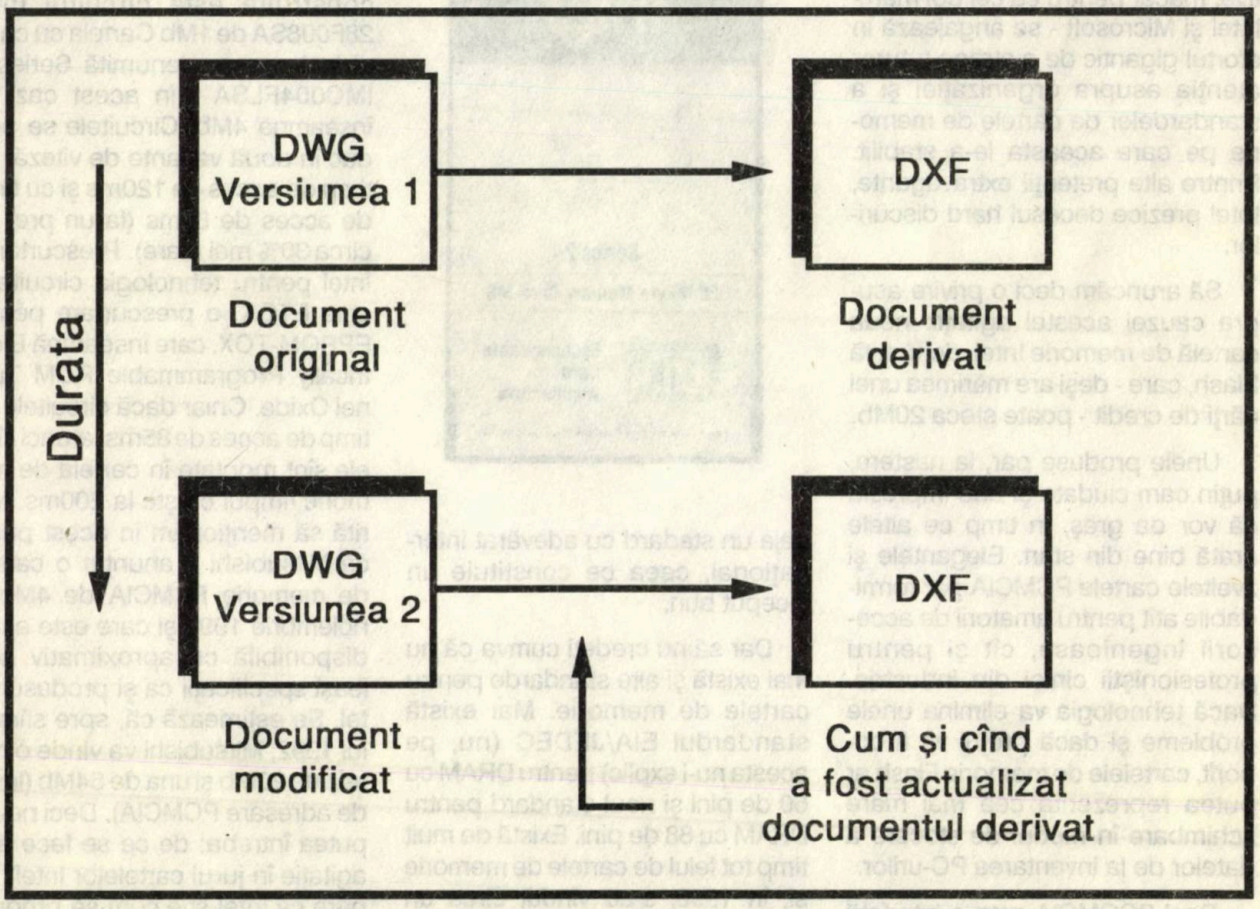


Diagrama 4: Într-o relaţie derivată, sistemul DMS determină modul în care un document derivă dintr-altul, cum ar fi, de exemplu, un fişier DXF obţinut dintr-un desen DWG realizat în AutoCAD. Într-un astfel de model, documentul derivat devine invalid dacă desenul original este modificat.

Flash stîrneşte senzaţie

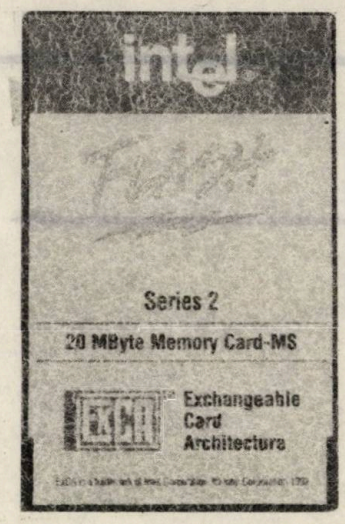
Cartelele de memorie Flash produse de Intel sînt compacte şi rapide. Datorită propriului lor standard internaţional şi datorită marelui interes manifestat de profesionişti, ele ar putea vesti sfîrşitul hard discurilor.

PCMCIA înseamnă Personal Computer Memory Card International Association (Asociaţia Internaţională pentru Cartele de Memorie pentru Calculatoare Personale). Despre această melodioasă abreviere veţi mai auzi multe în viitor, dacă nu din alte motive, măcar pentru că cei doi mari - Intel şi Microsoft - se angajează în efortul gigantic de a atrage tuturor atenţia asupra organizaţiei şi a standardelor de cartele de memorie pe care aceasta le-a stabilit. Printre alte pretenţii extravagante, Intel prezice decesul hard discurilor.

Să aruncăm deci o privire asupra cauzei acestei agitaţii: noua cartelă de memorie Intel, denumită Flash, care - deşi are mărimea unei cărţi de credit - poate stoca 20Mb.

Unele produse par, la naştere, puţin cam ciudate şi dau impresia că vor da greş, în timp ce altele arată bine din start. Elegantele şi zveltele cartele PCMCIA par formidabile atît pentru amatorii de accesorii ingenioase, cît şi pentru profesioniştii cinici din industrie. Dacă tehnologia va elimina unele probleme şi dacă preţul va fi coborît, cartelele de memorie Flash ar putea reprezenta cea mai mare schimbare în modul de stocare a datelor de la inventarea PC-urilor.

Deşi PCCMCIA este o interfaţă standard, standardele se modifică. PCMCIA este o organizaţie a unor fabricanţi cu aceleaşi intenţii care s-au întîlnit în S.U.A. în august 1989 pentru a se pune de acord cu privire la un standard pentru cartele de memorie. În acelaşi timp, un grup din Japonia (fiţi pregătiţi pentru alte iniţiale) numit JEIDA (Japaneze Electronics Industry Development Assocciation) are propriile sale idei. Deşi este uimitor pentru industria computerelor, cele două organizaţii au căzut de acord asupra unui standard comun în septembrie 1991, şi anume PCMCIA 2.0/JEIDA 4.0. Deci avem



deja un stadard cu adevărat internaţional, ceea ce constituie un început bun.

Dar să nu credeţi cumva că nu mai există şi alte standarde pentru cartele de memorie. Mai există standardul EIA/JEDEC (nu, pe acesta nu-l explic) pentru DRAM cu 60 de pini şi noul standard pentru DRAM cu 88 de pini. Există de mult timp tot felul de cartele de memorie şi, în 1990, s-au vîndut circa un milion. Entuziaştii PCMCIA (poate optimiştii) estimează ca vînzările să explodeze la 100 de milioane de unităţi în 1995.

Cartelele normale PCMCIA de Tip 1 au 3,3 mm x 85,64 mm x 54 mm. Cartelele de tip 2 mai puţin răspîndite, au grosimea de 5 mm şi celelalte dimensiuni identice. Pe una din laturile de 54 mm se găseşte un conector mamă cu 68 de pini (ambele cartele au grosimea de 3,3 mm pe această latură). De obicei, cartela se inserează într-un dispozitiv şi se împinge la locul ei. Pentru a o extrage, este nevoie de o sculă specială.

Noua cartelă de memorie Flash se livrează în trei variante: 4,10 şi 20MB. "Cărămida" cu care este construită este circuitul Intel 28F008SA de 1Mb Cartela cu chipul inclus este denumită Series 2 IMC004FLSA - în acest caz "4" înseamnă 4Mb. Circuitele se produc în două variante de viteză: cu timp de acces de 120ms şi cu timp de acces de 85ms (la un preţ cu circa 30% mai mare). Prescurtarea Intel pentru tehnologia circuitelor este ETOX, o prescurtare pentru EPROM-TOX, care înseamnă Electrically Programmable ROM Tunnel Oxide. Chiar dacă circuitele au timp de acces de 85ms, atunci cînd ele sînt montate în cartela de memorie timpul creşte la 200ms. Merită să menţionăm în acest punct că Mitsubishi a anunţat o cartelă de memorie PCMCIA de 4Mb în noiembrie 1991 şi care este acum disponibilă cu aproximativ aceleaşi specificaţii ca şi produsul Intel. Se estimează că, spre sfîrşitul lui 1992, Mitsubishi va vinde o cartelă de 20Mb şi una de 64Mb (limita de adresare PCMCIA). Deci ne-am putea întreba: de ce se face atîta agitaţie în jurul cartelelor Intel? Se pare că Intel ştie cum să producă senzaţie cu Flash.

Care este diferenţa între această memorie EPROM "Flash"

şi DRAM sau SRAM cu care sînt obişnuiţi utilizatorii de PC? Atît RAM-ul dinamic (DRAM) cît şi RAM-ul static (SRAM) au nevoie de energie pentru a-şi păstra conţinutul şi de aceea ele sînt denumite "volatile". De fapt, DRAM-ul are nevoie de operaţii constante de reîmprospătare ca şi un condensator cu pierderi.

În cazul cartelelor non - EPROM această energie este furnizată de o mică baterie. EPROM-ul nu are nevoie de energie pentru a păstra datele, motiv pentru care este denumit "nevolatil", şi este - în mod evident - util ca mediu de stocare. După cum ne aducem aminte, nici discurile nu au nevoie de energie pentru a păstra datele - dar ea le este necesară pentru citire sau scriere. Termenul "Flash" înseamnă că datele din circuitul EPROM pot fi modificate pe loc, spre deosebire de EPROM-ul uzual care trebuie să fie scos din calculator şi introdus într-un dispozitiv special pentru a fi reprogramat. Pe lîngă proprietatea de nevolatilitate, noile cartele de memorie - mai ales cele de 20Mb - sînt mult mai compacte decît uzualul DRAM SIMM care se găseşte în cele mai multe calculatoare.

În pofida numelui său, interfaţa PCMCIA propriu-zisă nu acceptă numai cartele de memorie. Mulţi producători au intuit posibilităţile acestui port care şi-a găsit locul în multe maşini şi va exista, se pare, în mult mai multe. Deja există pe piaţă cartele moderne PCMCIA, unul dintre producători fiind chiar Intel. Au mai fost ofertate cartele pentru reţea şi este în curs de apariţie chiar şi o interfaţă de imprimantă PCMCIA. Şi de ce nu? Acest tip de interfaţă a fost caracterizată ca "magistrala de calcul portabilă a viitorului". Nu mai este nevoie să spunem că Intel se află în frunte cu al său ExCa Exchage Card Architecture, un standard pentru drivere de dispozitiv pentru conectorul PCMCIA folosit, de exemplu, de modemuri. Este posbil ca prima întrebare pusă relativ la o nouă tehnologie să devină: "Se potriveşte cu o cartelă PCMCIA?".

Prezenţa lui Microsoft subliniază potenţialul şi viitorul progres ale noilor cartele Flash. Menţinîndu-se întotdeauna în zona profitului, gigantul a intrat în scena PCMCIA cu Flash File Systems versiunea 2.0 proiectat special pentru EPROM-ul Flash.

Deci, se pare că dispune de aceste minunate dispozitive de mărimea unei cartele de credit care are mari spaţii de stocare, care nu au nevoie de energie pentru a păstra datele, care sînt mici şi care pot fi scăpate pe jos fără riscuri. Ele pot face accesul la date de mii de ori mai rapid decît discurile, au un standard care să elimine redundanţele, şi beneficiază de implicarea lui Microsoft. Cu siguranţă că discurile hard au murit.

Nu este chiar aşa. Prima problemă o constituie preţul. Intel cere 369£ pentru o cartelă de 20Mb, ceea ce înseamnă 18,45£/Mb. Să menţionăm că trebuie să se cumpere mii de bucăţi pentru a le obţine la acest preţ. Ultimul disc Fujitsu de 3,5" in de 1082Mb cu timp de acces de 10ms şi rată de transfer de 10Mb/sec poate fi cumpărat cu circa 2000£, ceea ce înseamnă 1,85£/Mb - o zecime din preţ. Un disc cu tehnologie mai slabă dar mai ieftin, discul Seagate de 1,4Gb, revine la 1,30£/Mb. Deci, în locul discurilor, Flash este foarte scump. Este la fel de interesant să comparăm densitatea datelor cartelei PCMCIA - care pare un avantaj major - cu densitatea datelor pentru discuri. Cartela Intel de 20Mb oferă 1,3Mb/cm3 în timp ce discul Fujtsu, un dispozitiv cu înălţime standard, oferă 1,8Mb/cm3. Tehnologia veche cîştigă din nou.

Desigur, discurile consumă mai multă energie în funcţionare decît cartelele Flash, iar în cazul PC-urilor alimentate de la baterii consumul este un aspect crucial. Citirea de la cartele cere puţină energie, dar cînd se începe scrierea într-o cartelă EPROM Flash încep să apară problemele şi nu numai în domeniul consumului. Noile cartele Intel folosesc două tensiuni: 5V pentru citire şi 12V pentru scriere/ştergere. EPROM-ul trebuie şters înainte a fi înscris şi acest proces este atît consumator de energie (pînă la 90mA), cît şi foarte lent. Iar utilizarea înrăutăţeşte performanţele. Într-un caz defavorabil al unei cartele vechi (intens folosite) se poate aştepta circa trei minute pentru ştergerea şi scrierea a 4Mb; însăşi operaţiunea de ştergere şi citire produce uzura cartelei de memorie Flash. Ştergerea blocurilor de date din circuite poate să aibă loc cu scrierea în altă zonă dar situaţia se înrăutăţeşte ori de cîte ori se ciclează blocul.

Aici intervine Microsoft cu al său Flash File System. Problema "uzurii" a fost recunoscută de fabricanţi şi o mare cantiate de inteligenţă a fost utilizată pentru a ţine urma blocurilor folosite şi a numărului de ciclări. Această logică utilizează "algoritmi de fixare a nivelului de uzură" şi ar putea diferi de la un fabricant de cartele la altul.

Standardul PCMCIA presupune că la începutul cartelei se află un antet care conţine informaţii specifice cartelei şi care se numeşte CIS (Card Information Structure). CIS conţine cele mai importante detalii referitoare la tipul cartelei cum ar fi tipul circuitului (Flash, SRAM, etc), structura internă, modul de adresare (cum ar fi la octet şi acces aleator) şi alte informaţii pentru interfaţă. De asemenea, CIS mai poate conţine adresa unei zone opţionale de memorie cu "atribute" care pune la dispoziţie informaţii de fabricant şi mai amănunţite despre organizarea cartelei. Aceste information sînt concepute pentru a permite o utilizare cît mai bună a diverselor abordări folosite de diverşi producători care folosesc aceeaşi tehnologie de bază.

Dacă scumpele cartele Flash se uzează prin utilizare, cum se poate şti cînd sînt complet uzate? În momentul de faţă, tehnologia cartelei măsoară durata ciclului de ştergere/scriere şi decide dacă este prea lungă. Dacă nu s-a întîmplat nimic în intervalul de timp dat, procesul este oprit şi repetat. Dacă după "n" repetări nu se întîmplă nimic, cartela este "moartă". Se poate presupune că, cu mult înainte ca aceasta să se întîmple, cartela raportează către Microsoft File System care, la rînsul său, îl informează pe utilizator despre epuizarea şi necesitatea înlocuirii cartelei din conectorul "A".

Deşi cartelele de memorie Flash sînt considerate "periferice", ele au posibilitatea de a fi folosite ca ROM suplimentar (sînt prea lente pentru a fi folosite ca RAM suplimentar). În cartele pot fi încărcate aplicaţii şi se foloseşte o specificaţie cunoscută sub numele de XIP (eXecute In Place), conform căreia un program este rulat direct de pe cartelă - fără a-l încărca mai întîi în memoria PC-ului. Aceasta prezintă avantajul evident de a elibera RAM preţios dar mai înseamnă că şi conectorul PCMCIA este ocupat la o singură aplicaţie.

Deci vestesc aceste noi cartele Intel Flash PCMCIA sfîrşitul erei discurilor hard aşa cum le cunoaştem? Astăzi răspunsul este nu. Dar împotriva neajunsurilor temporare, aceste cartele au proprietăţi remarcabile şi - pentru anumite aplicaţii - de utilitate imediată. Şi, ceea ce este mai important, combinaţia între succesul de piaţă al cuplului Intel/Microsoft şi forţa industrială a lui Mitsubishi promite un viitor mare acestui mic dispozitiv electronic.

O privire asupra procesării

Over Drive

Circuitele pentru dublarea frecvenţei de ceas pot ridica viteza globală a PC-urilor bazate pe 486 cu aproape 100%. Vă prezentăm modul lor de funcţionare şi aspectele care trebuie avute în vedere.

Dacă sînteţi puţin nedumeriţi în legătură cu opţiunile de îmbunătăţire a procesoarelor, nu sînteţi singurul. În timp ce Intel cheltuieşte milioane de dolari pentru a evidenţia - în reclame - un soclu gol, iar cei care vînd sisteme laudă tot felul de maşini modulare şi care pot fi dezvoltate, nu sînt puţini utilizatorii de PC-uri care îşi sparg capul încercînd să-şi dea seama ce să caute.

Căutînd să crească puterea, Intel a încercat să reducă dimensiunile chip-ului 486 şi să mărească frecvenţa de ceas la peste 50MHz. Dar chiar dacă frecvenţa circuitului se poate apropia de 100 MHz, restul sistemului - chiar şi pinii care conectează circuitul la placă - se confurntă cu tot felul de probleme la 50MHz şi mai mult. De asemenea, eliminarea interferenţelor cu frecvenţele radio devine mult mai dificil de realizat la frecvenţe ridicate ale ceasului.

Cache - dacă se poate

Din fericire, memoria cache inclusă în chip-ul 486 oferă o soluţie care permite să se lucreze cu sistemul la frecvenţe de ceas mai mari. În cea mai mare parte a timpului, procesul plasează instrucţiunile şi datele de care are nevoie în memoria cache din chip. Deoarece memoria cache lucrează independent de restul sistemului, ea permite procesorului să lucreze la o frecvenţă de ceas mai mare decît a restului sistemului.

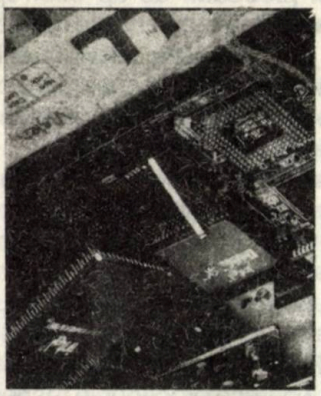
Pentru a exploata această posibilitate, Intel a creat o versiune de 486 cu un circuit de dublare a frecvenţei de 25MHz, îi permite să lucreze la 50MHz ori de cîte ori nu este necesar accesul la lumea exterioară. Intel va oferi aceste circuite pentru sistemele cu 486SX şi 486DX lucrînd la frecvenţe între 16MHz şi 33MHz.

Circuitul de dublare a frecvenţei ceasului promite o creştere a vitezei cu pînă la 100 la sută dacă aplicaţia este strîns legată de unitatea centrală şi dacă datele încap în memoria cache. Pe de altă parte, cu cît aplicaţia foloseşte mai mult discul, scrie mai mult la ecran sau utilizează mai des structuri de date care nu încap în memoria cache, cu atît va fi mai coborît cîştigul de viteză. Intel pretinde că utilizatorii aplicaţiilor populare de birou vor vedea îmbunătăţiri cu 62% pînă la 77%.

Circuitul de dublare a ceasului trebuie privit ca un 486 turbo. Circuitul se montează în acelaşi soclu ca şi 486 standard, iar semnalele de pe pini sînt aceleaşi. Diferenţa esenţială o constituie voiciunea pe care o dă sistemului: el lucrează în mod turbo - la o viteză dublă atunci cînd datele şi instrucţiunile necesare se găsesc în memoria cache.

DX2 şi OverDrive

Circuitele cu dublare a frecvenţei de ceas vin numai în versiuni 486DX - cu alte cuvinte, cu coprocesor matematic încorporat. Sînt disponibile două tipuri de circuite. Cel pentru fabricanţii de sisteme se numeşte 486DX2, iar cel pentru îmbunătăţiri instalate de către utilizatori se numeşte OverDrive. Cele două diferă numai prin configuraţia pinilor.



Timp de luni de zile Intel i-a tracasat pe utilizatori "scăpînd" cîte o informaţie în legătură cu circuitul de îmbunătăţire a performanţelor care se poate monta în soclul pentru coprocesor din sistemele cu 486SX. Firma se bazează pe acest produs pentru a-i lămuri pe utilizatori să renunţe la 386 şi să treacă spre 486.

Soclul pentru coprocesor din sistemele cu 486SX este într-adevăr un soclu pentru înlocuirea procesorului. Dacă se instalează un circuit în acest soclu, el va prelua rolul de procesor al sistemului, dezactivînd circuitul 486SX de pe placa de bază. Chip-ul 487SX, singurul circuit care poate fi instalat în acest soclu, este identic cu 486DX, cu excepţia unor mici diferenţe în asignarea pinilor.

Avantajele circuitului OverDrive

Dar un OverDrive este o alternativă mult mai tentantă decît un coprocesor. În timp ce 487SX îmbunătăţeşte viteza programelor - destul de puţine - care folosesc operaţii aritmetice în virgulă mobilă, circuitul OverDrive va mări viteza tuturor programelor.

Intel a introdus soclul de dezvoltare alături de 486SX pentru a permite îmbunătăţirea acestor sisteme. Dar şi sistemele cu 486DX includ un soclu similar. Nu ar avea sens să se instaleze un 487SX, deoarece s-ar înlocui circuitul 486DX de pe placa de bază cu un circuit identic.

Pe de altă parte, circuitul de dublare a frecvenţei de ceas constituie o îmbunătăţire atît pentru sistemele cu 486DX cît şi pentru cele cu 486SX. Pînă în acest moment, puţine sisteme cu 486DX conţin un soclu pentru OverDrive, dar majoritatea plăcilor de bază mai noi par să-l includă.

Deci ce opţiuni le sînt oferite celor care deţin sisteme cu 486DX fără socluri de dezvoltare?

Cea de-a doua versiune a circuitului 486 - turbo are aceeaşi configuraţie a pinilor ca un 486DX standard. În consecinţă orice sistem cu 486DX poate fi îmbunătăţit dacă se scoate circuitul 486DX şi se înlocuieşte cu un circuit de dublare a frecvenţei de ceas.

Necazurile cu garanţia

Această abordare ridică o problemă: nu este chiar atît de uşor să se scoată procesorul şi să se planteze un nou procesor pe placa de bază deoarece circuitul şi placa de bază se pot deteriora destul de uşor. Dacă se schimbă procesorul, producătorul sistemului va considera garanţia anulată. Oricum, mulţi utilizatori care posedă cunoştinţe tehnice îşi vor asuma riscul. Intel va produce "pachete" pentru îmbunătăţirea sistemelor cu 486DX care nu au soclu liber, dar va recomanda ca înlocuirea să fie făcută de fabricant, de cel care a vîndut sistemul sau de inginerul de service.

Se pot cumpăra circuite OverDrive de 25MHz (699$) şi de 16MHz sau 20MHz (549$) pentru PC-uri cu 486SX. Circuitele pentru utilizatorii de 486DX vor apare către sfîrşitul anului. Cei care cumpără sisteme cu 486DX2 dotate cu un soclu special de 238 de pini pentru OverDrive îşi vor putea îmbunătăţi maşinile utilizînd viitoarea generaţie de procesoare Intel care sînt codificate P5 - cîndva în 1993.

Michael Slater

Utilizarea mouse-ului în modul grafic 320x240x256

În acest articol vom face referire la alte două articole care au apărut în revista "If": în nr. 4/1991 (despre rutine scrise în Turbo Pascal pentru utilizarea mouse-ului), precum şi în nr. 4/1992 (despre programarea modului grafic 320x240x256).

În legătură cu utilizarea mouse-ului în acest mod grafic se impune o precizare: deoarece driver-ele de mouse nu recunosc acest mod (el nu este standard, singurul mod standard în 256 culori fiind 320x200), nu va fi posibilă afişarea cursorului grafic, şi deci nu vom putea utiliza nici o funcţie care se referă la desenarea cursorului grafic (definire cursor, afişare etc). Unit-ul *XMouse* vă pune la dispoziţie o colecţie de proceduri care să suplinească aceste lipsuri. Desigur, ele pot fi adaptate şi completate în conformitate cu necesităţile proprii.

Să explicăm acum fiecare procedură folosită în acest program. Înainte de a scrie un program propriu care să utilizeze aceste proceduri, citiţi cu atenţie indicaţiile de programare date în nr. 4/1991.

XResetMouse. Apelează funcţia 0 a driver-ului pentru reset hardware şi software. În plus se apelează funcţia 12 pentru a comunica faptul că dorim să controlăm orice eveniment de mouse (deplasare mouse, apăsare buton, eliberare buton). Procedura de tip Interrupt care realizează acest lucru este *ProcMouse*. Se precizează domeniul de valori pentru coordonatele X şi Y (pentru început am fixat valorile 0 - 319 pentru X, şi valorile 0 - 219 pentru Y; aceasta este rezoluţia pentru acest mod grafic). Cursorul de mouse este poziţionat în centrul domeniului (X = 160 şi Y = 120).

XModCursor. Precizează modul de desenare a cursorului. Modelul de cursor folosit în cadrul acestui program este memorat într-o matrice de 8x8 octeţi, iar punctul indicat de cursor ("hot spot") în două variabile de tip întreg. Înainte de desenarea cursorului într-o anumită poziţie pe ecran, porţiunea respectivă de imagine este salvată într-o zonă de manevră. Cursorul se desenează peste imaginea afişată pe ecran după următoarele reguli: dacă un octet din cadrul matricei are valoarea 0 se va păstra pixelul din imaginea veche (fundalul), altfel el va fi pus în poziţia respectivă. După această regulă putem să ne construim orice model de cursor folosind culori între 1 şi 255. De remarcat că această regulă de desenare a cursorului este mai simplă decît cea folosită de driver în modurile grafice standard.

ProcMouse şi *ProcAMouse*. Sînt procedurile care controlează orice eveniment de mouse. Driver-ului de mouse i se comunică (prin funcţia 12) faptul că procedura *ProcAMouse* asigură controlul. Această procedură asigură o funcţionare corectă pentru procedura *ProcMouse* (de tip Interrupt). Fără acest artificiu funcţionarea programului este posibilă numai dacă în sistem este instalat un driver A4Mouse (care lucrează corect cu două tipuri de periferice: A4 şi Genius). Acest driver permite să se conecteze la driver o procedură utilizator care se poate termina fie cu RETF, fie cu IRET. Alte driver-e (ex: Genius sau IBM) cer ca procedura să se termine obligatoriu cu RETF; evident, în această situaţie, conectarea unei proceduri de tip Interrupt conduce la blocarea sistemului în momentul în care ea este apelată. De altfel, documentaţia referitoare la programarea funcţiilor unui driver de mouse cere ca procedura să se termine cu RETF (observăm că driver-ul A4Mouse face în acest caz o concesie).

Acum este momentul să explicăm procedurile *XWritePixel* şi *XReadPixel*. În momentul apelului se verifică dacă coordonatele x şi y sînt în domeniul de valori permise (acţiune necesară deoarece este posibil ca uneori cursorul să nu fie vizibil în întregime). În plus secvenţele care operează cu regiştrii controlorului VGA sînt încadrate de PUSF/CLI şi POPF. Această protejare este necesară deoarece procedurile respective nu sînt reentrante (iar punctul critic îl constituie tocmai accesarea acestor regiştri). Prin această acţiune se permite ca rutina să fie întreruptă, iar procedura de tratare a întreruperii să apeleze tocmai rutina care fusese întreruptă, fără ca prin aceasta să fie afectată funcţiunea anterioară (care a fost executată doar parţial).

XShowCursor afişează cursorul în poziţia indicată de coordonatele memorate în variabila ZonSave, iar *XHideCursor* ascunde cursorul.

XPosCursor poziţionează cursorul de mouse într-un punct de coordonate precizate.

XTerMouse se apelează înainte de închiderea modului grafic, pentru ca driver-ul să deconecteze procedura utilizator *ProcMouse*.

Programul a fost testat pe mai multe calculatoare compatibile IBM-PC, cu mouse de tip Genius şi A4 (3 butoane), precum şi pe mai multe calculatoare IBM de tip PS/1 şi PS/2, cu mouse de tip IBM (2 butoane).

Precizări suplimentare

Programul prezentat lucrează cu o singură pagină de memorie ecran. În cazul în care se lucrează cu mai multe pagini, de regulă se doreşte ca afişarea cursorului de mouse să se facă în pagina de afişare (indiferent care este pagina de lucru). Pentru aceasta trebuie ca procedurile de setare a paginilor de afişare (respectiv de lucru) să fie plasate în unit-ul *XMouse* şi puse în

legătură cu celelalte proceduri. Schimbarea paginii de afişare se va face în acelaşi timp cu redesenarea cursorului (dacă este cazul). Procedura *XShowCursor* va seta corespunzător pagina de lucru înaintea afişării cursorului (deci în această situaţie pagina de lucru trebuie să coincidă cu pagina de afişare), iar după afişarea acestuia să restaureze pagina de lucru anterioară.

mat. Ioan Cozac

```
unit xmouse;
interface uses xgrafica,dos;
const dimcursor = 7;
type patterntype = record xcursor,ycursor:integer;
   buffer:array[0..dimcursor,0..dimcursor] of byte; end;
var leftb,rightb,centerb:boolean; zonsave:patterntype;

procedure xresetmouse;
procedure xmodcursor(pattern:patterntype);
procedure xshowcursor;
procedure xhidecursor;
procedure xposcursor(x,y:integer);
procedure xtermouse;

implementation
var model:patterntype; cstare,cstat:byte;
  regs:registers; activ:boolean;

procedure procamouse; far; external; {$l xmouse}

procedure procmouse
   (rfl,rcs,rip,rax,rbx,rcx,rdx,rsi,rdi,rds,res,rbp:word);
interrupt;
begin
   if not activ then
     begin
        activ: = true; cstat: = cstare;
        xhidecursor;
        zonsave.xcursor: = rcx;
        zonsave.ycursor: = rdx;
        if (rax and 2) < >0 then leftb: = true;
        if (rax and 4) < >0 then leftb: = false;
        if (rax and 8) < >0 then rightb: = true;
        if (rax and 16) < >0 then rightb: = false;
        if (rax and 32) < >0 then centerb: = true;
        if (rax and 64) < >0 then centerb: = false;
        if cstat0 then xshowcursor;
        activ: = false;
     end;
end;

procedure xresetmouse;
begin
   regs.ax: = 0; intr($33,regs);
   regs.ax: = 12; regs.cx: = 127;
   regs.es: = seg(procamouse);
   regs.dx: = ofs(procamouse);
   intr($33,regs);
   regs.ax: = 7; regs.cx: = 0; regs.dx: = 319;
   intr($33,regs);
   regs.ax: = 8; regs.cx: = 0; regs.dx: = 239;
   intr($33,regs);
   fillchar(zonsave,sizeof(zonsave),#0);
   with zonsave do
     begin
       xcursor: = 160; ycursor: = 120;
     end;
   regs.ax: = 4; regs.cx: = 160; regs.dx: = 120;
   intr($33,regs);
   cstare: = 0; activ: = false;
   leftb: = false; rightb: = false; centerb: = false;
end;

procedure xmodcursor(pattern:patterntype);
begin
   cstat: = cstare; xhidecursor; model: = pattern;
   if cstat0 then xshowcursor;
end;

procedure xshowcursor;
var i,j,xo,yo:integer; p:byte;
begin
   if cstare = 0 then
     begin
       xo: = zonsave.xcursor-model.xcursor;
       yo: = zonsave.ycursor-model.ycursor;
       for i: = 0 to dimcursor do
         for j: = 0 to dimcursor do
           zonsave.buffer[i,j]: = xreadpixel
                   (word(xo + i),word(yo + j));
       for i: = 0 to dimcursor do
         for j: = 0 to dimcursor do
           begin
             p: = model.buffer[i,j];
             if p0 then
               xwritepixel(word(xo + i),word(yo + j),p);
           end;
       cstare: = 1;
     end;
end;

procedure xhidecursor;
var i,j,xo,yo:integer;
begin
   if cstare0 then
     begin
       xo: = zonsave.xcursor-model.xcursor;
       yo: = zonsave.ycursor-model.ycursor;
       for i: = 0 to dimcursor do
         for j: = 0 to dimcursor do
           xwritepixel(word(xo + i),word(yo + j),
```

```
            zonsave.buffer[i,j]);
        cstare:=0;
    end;
end;

procedure xposcursor(x,y:integer);
begin
  if (x=0) and (x<320) and
    (y=0) and (y<240) then
    begin
      cstat:=cstare; xhidecursor;
      regs.ax:=4; regs.cx:=x; regs.dx:=y;
      intr($33,regs);
      zonsave.xcursor:=x;
      zonsave.ycursor:=y;
      if cstat<>0 then xshowcursor;
    end;
end;

procedure xtermouse;
begin
  regs.ax:=12; regs.cx:=0; intr($33,regs);
end;

end.
```

```
; XMOUSE . ASM
code  segment byte public
    assume cs:code
    extrn  procmouse:far
    public procamouse
procamouse    proc far
    pushf
    call   procmouse
    ret
procamouse    endp
code  ends
    end
```

```
program xutmouse;
uses xgrafica,xmouse,dos,crt;
const pattern:patterntype=(xcursor:4; ycursor:4;
   buffer:((0,0,6,6,6,6,0,0),
       (0,0,6,5,5,6,0,0),
       (6,6,6,5,5,6,6,6),
       (6,5,5,5,5,5,5,6),
       (6,5,5,5,5,5,5,6),
       (6,6,6,5,5,6,6,6),
       (0,0,6,5,5,6,0,0),
       (0,0,6,6,6,6,0,0)));
var i,j:integer;

begin
  xsetgraph;
  xsetcolor(1,0,0,63);
  xsetcolor(2,0,63,0);
  xsetcolor(4,63,0,0);
  xsetcolor(5,63,0,63);
  xsetcolor(6,63,63,0);

  for i:=0 to 319 do for j:=0 to 239 do
    xwritepixel(i,j,1);
  for i:=0 to 319 do
    begin
      for j:=0 to 1 do xwritepixel(i,j,2);
      for j:=119 to 120 do xwritepixel(i,j,2);
      for j:=238 to 239 do xwritepixel(i,j,2);
    end;
  for j:=0 to 239 do
    begin
      for i:=0 to 1 do xwritepixel(i,j,2);
      for i:=159 to 160 do xwritepixel(i,j,2);
      for i:=318 to 319 do xwritepixel(i,j,2);
    end;
  for i:=2 to 118 do
    for j:=0 to 2 do
      begin
        xwritepixel(i+j,i,4);
        xwritepixel(319-i-j,i,5);
      end;
  for i:=237 downto 121 do
    for j:=0 to 2 do
      begin
        xwritepixel(239-i+j,i,5);
        xwritepixel(i+j+78,i,4);
      end;

  xresetmouse;
  xmodcursor(pattern);
  xshowcursor;
  repeat
    if leftb then
      begin
        xhidecursor;
        repeat until not leftb;
        xposcursor(160,120);
        xshowcursor;
      end;
    if centerb then
      begin
        sound(440);
        repeat until not centerb;
        xposcursor(160,120);
        nosound;
      end;
  until rightb;
  xtermouse; xterminate;
end.
```

Copierea rapoartelor în PARADOX

O problemă în Paradox o reprezintă copierea unei specificaţii de raport de la un fişier la altul atunci cînd cele două fişiere nu au aceeaşi structură. Acest fapt devine foarte deranjant în momentul în care fişiereie au structuri foarte asemănătoare, aproape identice, şi copierea raportului nu este permisă.

Pentru înlăturarea acestui neajuns am scris următoarea procedură care are ca parametri de intrare numele fişierului sursă, numele raportului care se doreşte a fi copiat, numele fişierului destinaţie şi numele sub care va fi copiat raportul.

ing. Cristian Cosma

```
Proc Blinking(var,line)
 if len(var) > 75 then
    message "Length of the string too big !!"
    sleep 2000
    quit
 endif
cursor off
@ line,76-len(var) ?? " " + var
paintcanvas attribute 79 line,76-len(var),line,76
@ line,77 ?? "..."
paintcanvas attribute 207 line,77,line,79
endproc

Proc Copy.report(source,rs,target,rt)
private frs,nfs,nft,s1,s2,t1,t2,i,mm,com1,com2,j,k,jj,ss
reset clear
Blinking("Analysing source table and report
      specification existance",21)
sleep 800
if substr(type(rs),1,1) < > "A" then rs = strval(rs) endif
if substr(type(rt),1,1) < > "A" then rt = strval(rt) endif
if not istable(source) then clear beep
quit "Source table does not exist !!" endif
if not istable(target) then clear beep
quit "Target table does not exist !!" endif
if rs = "r" then
   frs = source + ".r"
else
   frs = source + ".r" + rs
endif
if not isfile(frs) then
clear
quit "Source report specification doesn't happen
      to be here !!"
endif
if rt = "r" then
   frs = target + ".r"
else
   frs = target + ".r" + rt
endif
if isfile(frs) then
   clear style attribute 79
   @ 4,12 ?? "Target report specification exists."
                + "Overwrite it ?"
   showmenu
      "No"    : "Leave the application",
      "Yes"   : "Overwrite the target report"
                + "specification and go on"
                + " with the application"
   to i
   clear
   if i < > "Yes"
      then Blinking("Leaving",21) sleep 800 quit
   endif
endif
clear Blinking("Creating dummy table",21)
nfs = nfields(source)  nft = nfields(target)
array s1[nfs]  array s2[nfs]  array t1[nft]  array t2[nft]
mm = min(nfs,nft)
array com1[nfs]  array com2[nfs]
create "X___h__q" like source
copyreport source rs "X___h__q" "1"
clear Blinking("Comparing tables",21)
clearall menu {Tools} {Info} {Structure}
   select "X___h__q"
 i = 1
scan
   s1[i] = [Field Name]  s2[i] = [Field Type]
   i = i + 1
endscan
clearall
menu {Tools} {Info} {Structure} select target
i = 1
scan
   t1[i] = [Field Name]  t2[i] = [Field Type]
   i = i + 1
endscan
k = 1
for i from 1 to nfs
   for j from 1 to nft
      if s1[i] = t1[j] and s2[i] = t2[j] then
         com1[k] = s1[i]  com2[k] = s2[i]  k = k + 1
         quitloop
      endif
   endfor
endfor
clearall
if k = 1 then
   quit "The source and destination tables don't have"
        + "any common fields !!"
else
    mm = k-1
endif
```

```
clear Blinking("Restructuring dummy table",21)
menu {Modify} {Restructure} select "X__h__q"
if istable("Problems") = True
   or istable("Keyviol") = True then {OK}
endif
ss = nrecords("Struct")
for ij from 1 to ss
   k = true
   for i from 1 to mm
      if [FieldName] = com1[i] and [FieldType] = com2[i]
         then k = false quitloop endif
   endfor
   home
for ij from 1 to nft
   j = recno()
   if [FieldName] < > t1[j] or [FieldType < > t2[j]
      then ins moveto [FieldName]
         ctrlbackspace typein t1[j] right
         ctrlbackspace typein t1[j] right
   endif
   down
endfor
do_it!
while menuchoice() = "Delete"
   {Delete}
endwhile
clearall
clear Blinking("Copying desired report",21)
copyreport "X__h__q" "1" target rt
delete "X__h__q"
reset clear beep
quit "It's been done succesfully !!"
endproc
```

Reprezentarea grafică a unei funcţii

"surface.pas" realizează reprezentarea unor funcţii de forma z=f(x,y). Versiunea actuală reprezintă grafic funcţia

f(x,y)=cos(x2+y2)

în limitele intervalului

- S*L <= x, y <= S*L

cu un pas de S=0,15 şi un număr de paşi L=28.

Aici se renunţă la algoritmii Hidden-Line de altfel uzuali; în locul acestora folosiţi sub Turbo Pascal procedura Fillpoly. Dacă în linia

Setfillstyle (1,...)

înlocuiţi 1 cu 0, atunci desenaţi în locul suprafeţei un model în reţea.

Reprezentări "frumoase" vor da, de exemplu, funcţiile:

f(x,y) = exp(-x2-y2)
f(x,y) = sin(x) + cos(y)
f(x,y) = 1,5xyexp (sin(x) + cos(y))/(1 + x2 + y2)
f(x,y) = 1,5cos (3(abs(x)-abs(y)))

Eventual trebuie să modificaţi atunci parametrii de interval S şi L, ca poza să încapă complet în ecran. Plasarea punctului privitorului şi pregătirea, o manipulaţi prin parametrii de privire R, D, T, şi P. Deci nu se pun bariere fanteziei.

Programul îl puteţi compila cel puţin începînd cu versiunea 5.0. Aveţi grijă aici, să puneţi corect calea spre driver-ele BGI în procedura *Initgraph*. Programul rulează cu adaptoarele grafice uzuale Hercules, EGA şi VGA. Prin opţiunea de compilare de la începutul programului, utilizaţi şi un coprocesor.

Dacă doriţi să introduceţi interactiv funcţia, este de folos următorul truc:

1. Împărţiţi programul în 3 părţi:

- partea 1: "inceput.pas" lucrează pînă la

"function f(...):real;"

- partea a 2-a: "mijloc.pas" foloseşte o funcţie (vezi punctul 2)

- partea a 3-a: "sfîrşit.pas" prelucrează restul, începînd cu "end;" inclusiv.

2. Citiţi funcţia (de exemplu prin program Pascal sau prin comandă DOS "copy con mijloc.pas") ca şir de caractere f: = cos(sqr(x) + sqr(y))

de exemplu într-un fişier "mijloc.pas"

3. Puneţi cele trei părţi din nou împrejur prin

copy început.pas+mijloc.pas+sfirşit.pas

4. traduceţi "f.pas" prin "tpc f" pentru programul executabil.

"tpc.exe" este în cele ce urmează versiunea cu linii de comandă a compilatorului Pascal. Instrucţiunile din linia 7 puteţi să le preluaţi şi în compilatorul cu linii de comandă, la care "tpc.exe" îl comandaţi cu următoarea sintaxă:

tpc f/$r-/$s-

Linia 8 vă prezintă procedeul compilării condiţionate. Aici definiţia "IFDEF" cu comutatorul "CPU87" se referă la computere care utilizează un coprocesor.

5. Apelaţi "f.exe"

Aici paşii 2 pînă la 5 trebuie să permită rezolvarea unui fişier batch, la care eventualele gerşeli sînt interceptate prin variabila "Errorlevel". Pentru a creşte viteza, toate aceste operaţii ar trebui să lucreze pe un RAM - disc. În continuare parametri de interval şi de privire/perspectivă pot fi, de asemenea, fixaţi interactiv; la fel puteţi desena, printr-o simplă interogare în program, în locul unei suprafeţe, modelul în reţea.

(Matthias Fisches/et)

```
{Program: surface.pas}
{$S-,R}
{$IFDEF CPU87} {$N+}{$ELSE}{$N+,E+}
{$ENDIF}
uses Crt,Graph;
  nst   r=20;
        d=900;
        t=0.5;
        p=1.1;
        s=0.15;
        l=28;
type real=single;
var   g,h,i,j,k,xc,yc: Integer;
      x,y,st,sp,ct,cp: real;
      v: array[1..4]of pointtype;

function f(x,y: real): real;
begin
   f:=Cos(Sqr(x)+Sqr(y))
end;
procedure q(var p: pointtype;x,y,z: real);
var a,b: real;
begin
   a:=-(x*ct+y*st);
   b:=(a*sp-z*cp+r)/d;
   p.x:=Round(xc+(y*ct-x*st)/b);
   p.y:=Round(yc-(a*cp+z*sp)/b)
end;
begin
   g:=0;
   InitGraph(g,h,'');
   if GraphResult<>0 then Halt;
   xc:=GetMaxX SHR 1;
   yc:=GetMaxY SHR 1-30;
   SetFillStyle(9,1);
   Bar(0,0,GetMaxX,GetMaxY);
   SetFillStyle(1,4*Byte(g7));
   st:=Sin(t); sp:=Sin(p);
   ct:=Cos(t); cp:=Cos(p);
   for I:=-l to l-1 do
   for j:=-l to l-1 do
   begin
      for k:=1 to 4 do
      begin
         x:=(I+k AND 2 SHR 1)*s;
         y:=(j+(k-1)SHR 1)*s;
         q(v[k],x,y,f(x,y))
      end;
      FillPoly(4,v)
   end;
   Rectangle(0,0,GetMaxX,GetMaxY);
   repeat
   until KeyPressed;
   CloseGraph
end
```

EGA cu 25/43 linii

Vă prounem în continuare două scurte programe care permit comutarea ecranului (conectat la o interfaţă EGA) în mod 43 linii x 80 coloane şi înapoi în modul 25 linii x 80 coloane.

Programele pot fi lesne testate cu ajutorul utilitarului DEBUG. Listingurile sînt sub forma unor "fişiere de comenzi". Lansaţi DEBUG cu linia:

```
debug < line43.deb
```

C.N.

```
; LINE43.COM
a
mov     ax,1112
int     10
mov     ax,0100
mov     cx,0406
int     10
mov     ax,1200
mov     bl,20
int     10
mov     ax,4c00
int   21

rcx
19
nline43.com
w
q
```

```
; LINE25.COM
a
mov     ax,0003
int     10
mov     ax,0500
int     10
mov     ax,1111
mov     bl,00
int     10
mov     ah,01
mov     cx,0809
int     10
mov     ax,4c00
int     21
rcx
1d
nline43.com
w
q
```

Cod	Nume pachet	Nr. dischete	For-mat	Capa-citate	Lim-ba
39	BALPRO - balanţă contabilă	1	5,25"	360k	R
40	UNO - completare formulare	1	5,25"	360k	R
45	Hamor Soft: hMISS - secretariat, hCONT-contabilitate, hGEN&UTIL - dezv., aplicaţii	1	5,25"	360k	R
47	Comprex S.R.L.: Spell check pentru limba română	1	5,25"	360k	R
46	AvantGarde Soft: Personal Database - secretariat & contabilitate	1	5,25"	1,2M	R
48	Spero Menus 1.0 - bibliotecă de meniuri orientată obiect	1	5,25"	1,2M	R

SHARE - if

În lista de mai sus sînt trecute programe demonstrative pentru mai multe produse software. Aceste programe "demo" pot fi obţinute prin comandă. Taxa pentru copierea unei dischete este de 50 lei.

Pentru toate programele de mai sus vă putem indica numele şi adresa autorilor.

PROPUNERE

Sînt convins că în România există mulţi informaticieni "solitari" şi multe firme cu un număr redus de angajaţi care, prin eforturi remarcabile, au elaborat programe ingenioase şi valoroase. Unele din aceste programe, nu foarte mari ca dimensiune, pot fi extrem de interesante pentru mulţi dintre noi. Şi, mai ales, pot constitui soluţia ideală pentru surmontarea unor dificultăţi întîlnite de mulţi alţi colegi de breaslă. Dar cum ar putea fiecare să afle de existenţa celuilalt?

Pentru a facilita aceste contacte şi pentru a contribui la activarea pieţei de soft de pe la noi (totuşi, cred că "piaţă de soft" nu înseamnă numai AutoCAD sau FoxPro sau Corel Draw sau OS/2) ne-am gîndit să înfiinţăm un fel de serviciu de distribuire a programelor.

Avînd în vedere costurile unei campanii publicitare, este imposibil pentru un student sau pentru un grup de 2-3 "softişti" să-şi promoveze programul de protecţie prin parolă a discului hard sau pe cel de balanţă contabilă sau pe cel de editare simultană a două fişiere text.

De aceea vă facem următoarea propunere: folosiţi-vă de revista "if" pentru a promova şi a vinde programele pe care dumneavoastră le consideraţi interesante. Revista ar putea asigura un spaţiu de prezentare a programelor (putem să-i spunem şi reclamă) şi expedierea dischetelor cu programe şi documentaţii către solicitanţi. Pentru aceste servicii, revista şi-ar rezerva un comision de 20-25% din preţul de vînzare propus de autor.

Evident, fiind vorba de o propunere, cifrele şi metodele de lucru sînt "negociabile".

Dacă ideea vă stîrneşte entuziasmul (sau măcar interesul) nu ezitaţi să ne contactaţi !

Redacţia

O REVOLUŢIE

în lumea reţelelor de calculatoare

LANtastic, produs ARTISOFT - SUA

La un preţ de cîteva ori mai mic

decît al reţelelelor clasice, obţineţi

aceleaşi funcţii esenţiale şi o

CALITATE GARANTATĂ 5 ANI

Pentru a primi relaţii şi

materiale documentare

căutaţi-ne chiar acum la

TEL / FAX 99 / 436-433

ART COMM GROUP

Distribuitori / intermediari

zonali sînt bineveniţi

Reţelele LANtastic vă oferă ocazia realizării visului oricărui posesor de calculatoare, cu menţinerea tuturor facilităţilor oferite de alte firme, cu mai multă flexibilitate şi la un preţ incomparabil mai mic.

NU EZITAŢI. ACESTEA SÎNT REŢELELE PE CARE LE-AŢI AŞTEPTAT PÎNĂ ACUM!

str. Gheorghe Doja 36
Tîrgu-Mureş ROMÂNIA
Adresa poştală: C.P. 64, O.P. 1
4300 Tîrgu-Mureş
ROMÂNIA
Telefon/Telefax: 004-954-31660

AUTHORIZED REMARKETER
- IBM PERSONAL COMPUTER PRODUCTS

AUTHORIZED REMARKETER
- IBM PERSONAL SYSTEM/1 COMPUTER

vă oferă

CALCULATOARE PERSONALE

din gama PS/1 şi PS/2

Raportul preţ/performanţă
este în favoarea Dvs. !